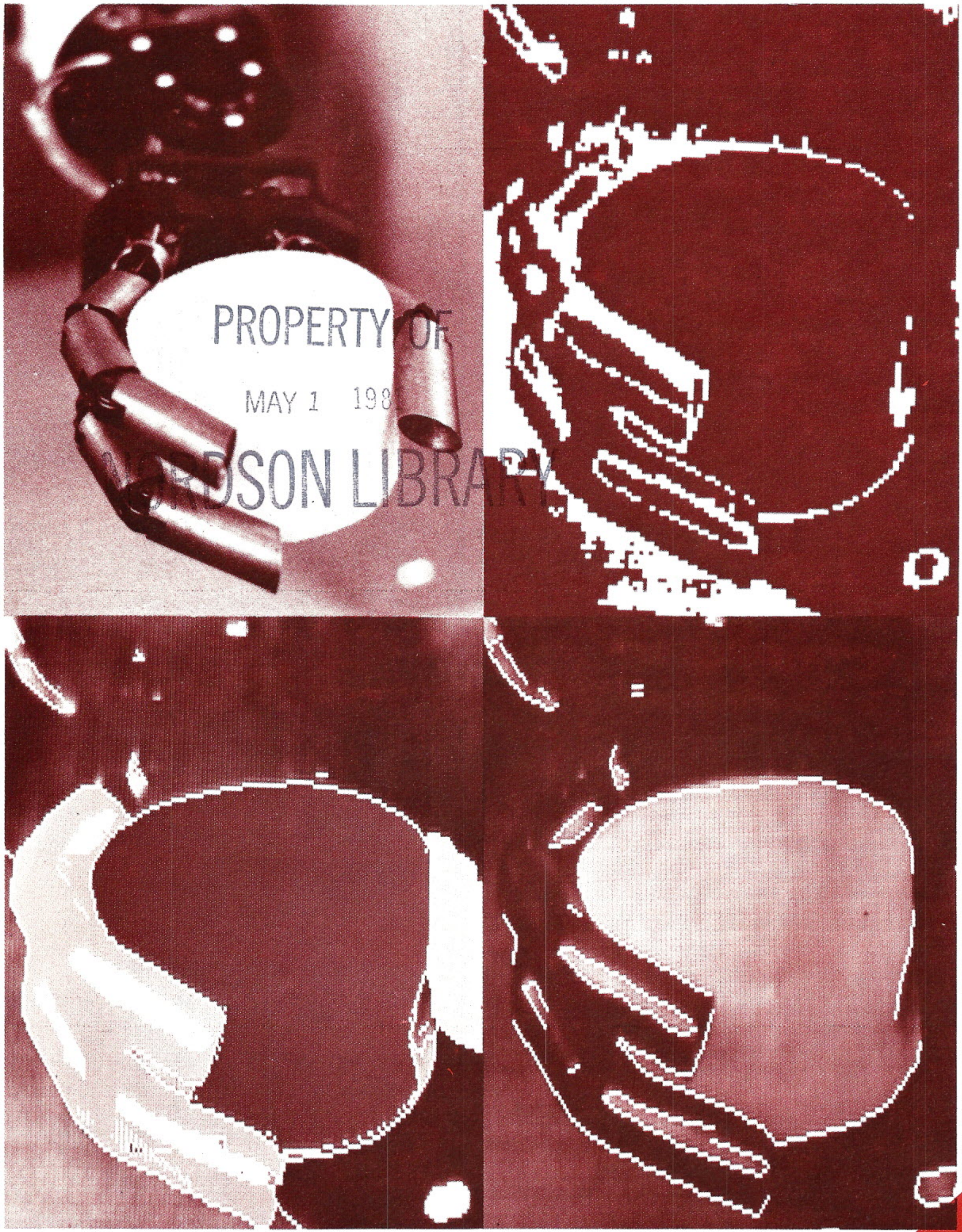
ROBOTICS AGE™

The Journal of Intelligent Machines™ • MAR/APR 1981 • $3.00 USA/$3.50 CANADA



SPECIAL ISSUE: Microcomputer Vision

SPECIAL ISSUE:

Microcomputer Vision

Robot vision is rapidly making its way out of the laboratory and into practical application. Although university researchers are creating ever more powerful image-interpretation systems on large machines, it's time for other experimenters to adapt some of the proven methods for use in microcomputer robot systems.

In this special issue, *Robotics Age* examines practical vision hardware and software techniques—presenting them to the general public for the first time. The issue leads off with a design for a inexpensive digitizer circuit, one that converts a standard TV camera signal to a silhouette-like image in memory. You need only a DMA interface to use it, and a TV camera that runs off synchronizing signals provided by the digitizer.

Next up is an explanation of *Chain-Code*, a technique you can use to compress the boundary information produced by the digitizer. Its uses extend beyond saving storage, however. With chain-code, you can compute the vital statistics of a closed boundary to locate its centroid, find its orientation, and describe its shape for comparison with known types. We've provided all the juicy mathematical formulas for this process—so you can implement it on your own computer.

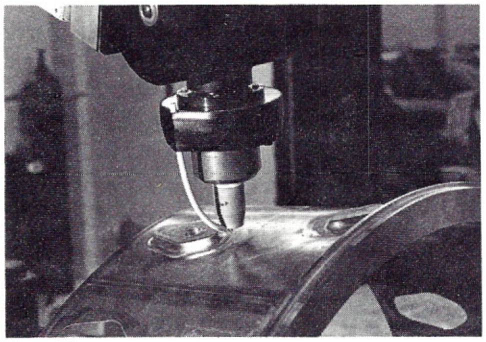
Relating an object's position in an image to its location outside the robot is not a trivial task, as explained in our third article, on *camera geometry*. But once you know the math behind the imaging process, as well as how to calibrate the camera/digitizer combination, you'll have the basic ingredients for effective target ranging, as you'll soon see.

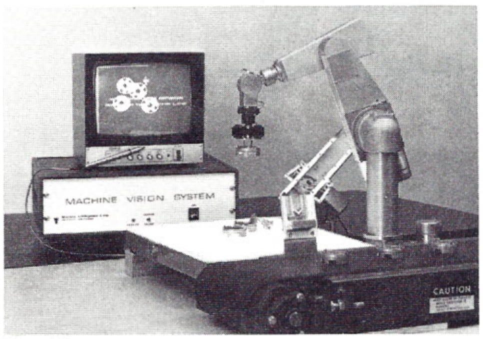
Look for more on practical robot vision and current vision research in future issues of *Robotics Age*, and let us know how these articles, as well as the other features in this issue, meet your interest.

—Alan Thompson

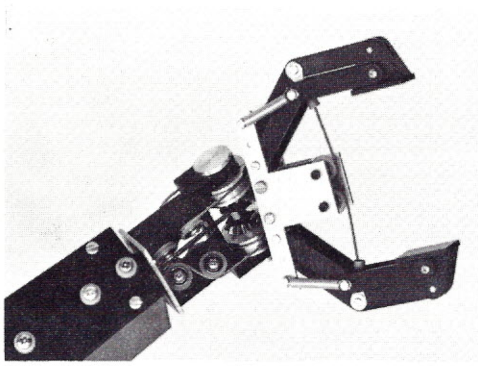
Our Cover shows three steps in the computer vision process. Starting clockwise, you'll find the original photograph of a robot hand; a bi-thresholded image, the kind produced by the digitizer circuit; an image with superimposed edges, such as those described in "Chain-Code"; and an image partitioned into closed regions by a higher level process.

ROBOTICS AGE MAR/APR 1981 Vol 3., No. 2


Page 28



Page 40


Page 41

Articles

Departments

Publisher
Philip C. Flora

Editorial Director
Alan M. Thompson

Managing Editor
Joel M. Wilf

Industrial Editor
J. W. Saveriano

Art Director
Robert Tinney

Production Editor
Cindee Huff

Technical Art
Heather Marr
James Marr

Graphics
Steven Fagerquist

Advertising
Robotics Publishing Corp.
10049 Commerce Ave.
Tujunga, Calif. 91042
(213)352-7937

ROBOTICS AGE—USPS 523850 (ISSN 0197-1905) is published six times a year, in January, March, May, July, September, and November, by Robotics Publishing Corporation, 10049 Commerce Ave., Tujunga, California 91042. Subscriptions are $15.00 for one year in the USA. Second-class Postage paid at La Canada, CA 91011. **POSTMASTER:** Send address changes to P. O. Box 801, La Canada, CA 91011.
Subscriptions from Canada and Mexico are $17 for one year. Other foreign countries $19 for one year surface delivery. Single copy price is $3.00 in the U.S. and $3.50 in Canada and Mexico, $4 in Europe, and $4.50 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a US bank. **Address all correspondence to:** Robotics Age, P. O. Box 801, La Canada, CA 91011, phone 213/352-7937. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of ROBOTICS AGE.

A TV digitizer that reads in binary images at video frame rates.

VIDEO SIGNAL INPUT

Introduction

Television cameras have the potential to provide for a robot what eyes provide for a human: the richest source of sensory information. But before a computer can get useful information from a camera, the analog video signal must be converted to a digital representation in the computer's memory. This is the function of a video image *digitizer*.

The conversion process begins by partitioning the image into a rectangular array of picture elements, or *pixels*. At each pixel, the digitizer measures the average brightness, assigns an integer *gray-level* that corresponds to that brightness, and stores it in computer memory. The digitized image, therefore, is typically stored as a two-dimensional array of gray-levels. Each gray-level's location in the array—its row and column position—corresponds to the pixel's location in the image (or is directly related to it). Figure 1 illustrates this scheme.

In this article I will describe the design of an inexpensive digitizer. For about $50 in parts, it will interface a television camera to a standard microcomputer DMA (direct memory access) port and store an array in the computer's memory that is derived from the video image. First, though, let's explore some of the considerations that influence the design.

Digitizer Design Considerations

The spatial *resolution* of a digitized image is related to the number of pixels in each image row and column defined by the digitizer. The theoretical resolution of an image refers to the volume of information available. When more pixels are sampled in an image, the image is usually represented with more accuracy, up to the theoretical limit. 35mm motion picture film, with a resolution of about 2000 × 2000, has a far greater image quality than standard broadcast television, whose resolution is on the order of 480 × 320 pixels. The human eye detects an image with a resolution of more than 4000 × 4000 pixels.

Clearly, increasing the spatial resolution of the digitizer beyond the theoretical resolution of the camera provides no increase in image information. Image digitizers that operate by scanning film can be designed with a great deal of flexibility, as the spatial resolution can be set rather arbitrarily. For a video digitizer, however, this is not the case, since the serial signal transmission requirements dictate that the image must be partitioned into rows (lines) at the camera. Thus, our row resolution is limited by the scan circuitry in the camera and is either difficult or impossible to change.

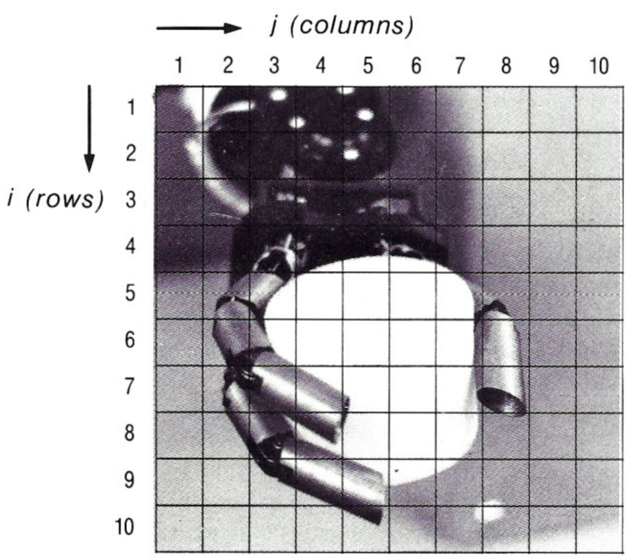
The horizontal (column) resolution of a video signal, however, is determined by the quality of the camera. More expensive cameras have a higher data rate, or bandwidth. Studio cameras, costing thousands of dollars, have the greatest bandwidth. An inexpensive monochromatic camera for home video or closed circuit television (CCTV) surveillance typically has less than a 3 MHz bandwidth, for which a 256 pixel column resolution is reasonable. Between these extremes lie numerous other types, including the commercially available solid-state CCD or CID cameras whose resolution is fixed by the lattice of light-integrating cells in the silicon substrate.

These solid-state cameras, by eliminating the analog drift intrinsic in vidicon scan circuitry, provide the most accurate calibration for automatic range measurement. However, their relatively high cost ($2000-5000), places them beyond the budget of most schools and individual experimenters. For our design, therefore, we will assume that a cheap CCTV camera will be used, and select a column resolution of 256 pixels per line.

The Electronics Industries Association (EIA) has defined a standard format for video signals used in the U.S., designated RS-170. In this standard, the image has 480 lines and is divided into two *fields*—each consisting of 240 lines scanned from top to bottom. One field contains all the odd numbered lines; the other field contains the even numbered lines. Each field is scanned in 1/60 second, and scanning is alternated between fields. Between each field, the scan has several line periods to return to the top, bringing the total number of horizontal cycles per complete vertical cycle to 525. If we sample one complete field, the spatial resolution of the image will be 240 × 256 pixels. Figure 2 diagrams the EIA video signal format.

Gray-scale resolution is another important characteristic of the digitized image. This is the number of possible brightness values that can be assigned to any pixel. In many image processing applications, an 8 bit gray-scale is used, for a resolution of 256, where zero represents a completely dark pixel and 255 represents one of maximum brightness. With one byte of storage required per pixel, a 240 x 256 pixel image requires 61,440 bytes of computer memory—practically enough to consume the 16 bit address space of most micros!

One approach that is becoming more economical as memory prices come down is to dedicate an entire separate address space to storing the image. With the memory alternately accessible from the computer and the



j (columns)

i (rows)	1	2	3	4	5	6	7	8	9	10
1	45	78	35	34	41	40	74	121	126	128
2	42	67	12	27	31	44	66	115	129	124
3	40	63	28	25	20	27	59	109	117	126
4	62	74	49	110	136	145	138	102	112	115
5	124	103	77	226	238	243	240	104	109	191
6	127	117	69	109	221	230	225	109	101	107
7	130	127	66	78	118	210	205	93	85	103
8	134	134	85	73	129	175	190	125	99	100
9	138	135	133	199	80	94	98	104	105	108
10	139	136	133	129	126	122	112	119	139	171

Figure 1. The first step of digitization is to partition the image into cells (pixels) addressed by row and column, shown (exaggerated) on the left. Within each pixel, the digitizer measures and assigns a number corresponding to the image brightness (right).

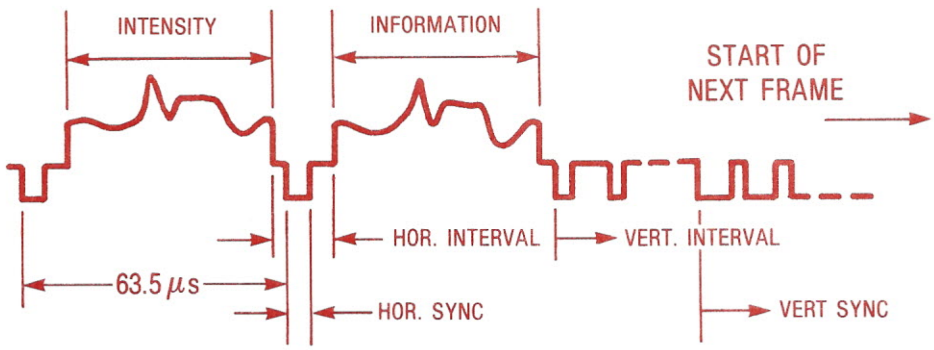


Figure 2. An EIA RS-170 composite video signal. The period between successive horizontal intervals represents one horizontal scan line in the image. The start of the horizontal interval terminates the intensity information on the line, and the horizontal sync pulse triggers a "flyback" to the start of the next line down. A full frame of lines is terminated by the start of the vertical interval, and vertical sync causes the scan of the next frame to begin at the top.

digitizer, the latter can deposit a digitized image in the memory at the full video rate, later making individual pixels available to the computer at memory access speeds. Because of its similarity to taking a "snapshot" of the video scan, this technique is called *frame grabbing*. Many types of frame grabbers are commercially available, some even for the S-100 bus (the CAT-100 by Digital Video Systems, for example), but the prices are still relatively high ($1000-$10,000), and constructing one from parts would be an ambitious project for the amateur.

Fortunately, there are many ways we can reduce the amount of image memory needed, making it feasible to use the computer's main memory for image storage. We could, for example, store only a portion of the image. With two pixel addresses, we can define the upper left and lower right corners of a rectangular region in the image. If this "window" is programmable, we can read only the rectangular portion of the image that contains useful information or that we have room to store and process.

We can further reduce the amount of memory the image needs by sacrificing spatial resolution. But this throws away valuable information—for many applications it would be quite useful to have the full 240 x 256 pixel resolution. It makes more sense to lower gray-scale resolution. If we allow each pixel only two possible gray-levels (0 or 1), we reduce our storage needs eightfold, to less than 8k bytes. Using this "binary image" has the added advantage of making our analog-to-digital conversion (ADC) circuitry much simpler and less expensive.

In addition to the problem of allocating memory for an entire image, the video input system must deal with the high data rate, or bandwidth, of a video signal. If we want to read in a 256 pixel line at the full video rate, we would need a data rate of over 5MHz—beyond the capability of most microcomputers and high enough to effectively stall computation on most minis. By accepting a lower gray-scale resolution and buffering several pixels per data word—in our case 8 per byte—we do not need as high a data rate.

If our camera were looking at an unchanging scene, we would not have to grab the whole image in 1/30 second. We could lower the data rate by sacrificing digitizing time rather than resolution. There are two common ways of doing this: with *point* and *column* digitizers. A point digitizer keeps track of the pixel row and column address in the scan, and when it matches the address of the desired point, it samples the video signal, converts it to a gray level, and presents the result to the computer. The computer may have to wait up to 1/30 second for the desired point to "come around" in the scan. This is the scheme used in the "Digisector," by Microworks, Inc.

A column digitizer is able to read in an image much faster. Given the number of a desired column, it digitizes the pixel for that column on each successive line in the image. At the end of 1/30 second, when every line has been scanned, a complete column of pixels has been stored in memory. Reading only 30 columns/sec (max.), a column digitizer would take over 8 seconds to convert an entire 256 column image. Since a robot vision system usually has to react to a changing environment, I chose to discard both these low-speed input schemes in favor of frame grabbing—reading the image in one 1/30 second scan, but with only one bit per pixel to reduce the data rate and storage requirements.

We can design the digitizer so that it grabs one video field of 240 x 256 pixels every second and stores it as a binary image in memory. Adding a few more requirements, though, would make it a much more powerful device. First, the actual data transfer should be performed by a standard DMA card. For image transfer using this digitizer design on an 8 bit machine, the DMA must be able to handle one byte every 1.6 microseconds. A 16 bit machine would need to handle only a 3.2 microsecond DMA cycle.

Another attractive feature is *windowing*. Sometimes only a portion of an image is worth considering. The digitizer should be able to select any rectangular window in the image, and read in only that part. The window should be set under computer control.

Variable thresholding is another function that would

improve the digitizer's power. Instead of having just one fixed analog level as the boundary between the digitized values of 0 and 1, we would like to set an arbitrary level under computer control. Furthermore, we can extract more information from the picture by setting *two* thresholds and having the digitizer output a "1" only if the brightness falls *between* them and "0" otherwise. I have provided this feature, in addition to windowing, in the digitizer design presented here.

The Question of Synchronization

Most commonly available CCTV cameras produce a *composite* video signal (Figure 2) which contains the image intensity information as well as the horizontal and vertical sync signals that terminate a line and field, respectively. This poses a problem in the design of a video digitizer, since the digitizer must use these signals separately to reset its row and column counters. A sync separator circuit could be added to extract these signals (and this is the approach used in some commercially available systems), but these circuits are not exactly simple, and adding them increases both the cost of the digitizer and the effort required to build it. They also do not solve the other problem that arises when we use composite sync from the camera—the accuracy of the pixel address.

The horizontal and vertical oscillators used in inexpensive video cameras are usually entirely analog, and therefore subject to drift and various other instabilities. Attempting to synchronize the pixel clock, with a frequency 256 or more times the horizontal rate, with such an external horizontal sync reference will result in the column addresses near the end of a line becoming increasingly less accurate, with concomitant loss of accuracy when these addresses are used to compute the 3-space coordinates of targets from their position in an image. This inaccuracy can occur, though to a lesser extent, even if a solid-state camera with geometrically precise pixel locations is used—unless you have access to the same pixel clock used by the camera (which fortunately is the case with most of these cameras).

A much better solution, and the one selected for use in this digitizer, is to produce all video timing signals, both for the camera and the digitizer, from a single crystal-controlled master pixel clock. The only problem with this approach is that you must either obtain a camera that is capable of running on externally provided sync signals or get the schematics and modify one so it can. The work involved in modifying one is actually much easier than building a sync separator and results in a much cleaner and more accurate system.

The key to this approach is that there are inexpensive (about $10) TV sync generator ICs readily available from numerous sources that generate all necessary video timing signals. The one I selected for this circuit is the MM5320N (or the MM5321), by National Semiconductor. Clocked at a multiple of the pixel clock frequency, this chip produces either composite sync or separate horizontal and vertical drive signals for your camera, plus some other essential information.

One important signal that the 5320 gives the digitizer tells it where to look on the video signal for the DC level that corresponds to black. Since video signals are usually capacitively coupled, you cannot count on this "black" level to be at the signal ground, where it should be for the digitizer to make accurate conversions. As part of the RS-170 video signal standard, the EIA has specified that a portion of the signal immediately following the horizontal sync pulse should provide the black DC reference. The 5320 provides a signal that is active only during this interval, called the "Color Burst Gate" (CBG), since the interval may also contain color reference information. Of course, we shall be concerned here only with black and white signals.

The Digitizer Design

Given all the design considerations, we are ready to look at the functional architecture of the digitizer, shown in Figure 3. The pixel clock, shown in the lower left, provides a reference for the sync generator and also serves as the clock for the pixel column counter and for the shift register that buffers eight pixel values prior to their output to the computer. The signals produced by the sync generator are transmitted to the camera by 75ohm coaxial line drivers. Horizontal and vertical drive signals are also used to preset the column and row counters, respectively.

Synchronized with the reference drive signals, the camera returns its video output to the digitizer. The CBG signal from the sync generator triggers a DC restoration circuit that clamps the black reference to ground, resulting in a normalized analog video signal with image intensity levels across a line ranging from 0 to a maximum of .7 Volts. This referenced video is made available to the threshold comparators for conversion to binary gray levels.

The threshold levels used in the comparisons are set by the computer. Eight data lines from the CPU bus are

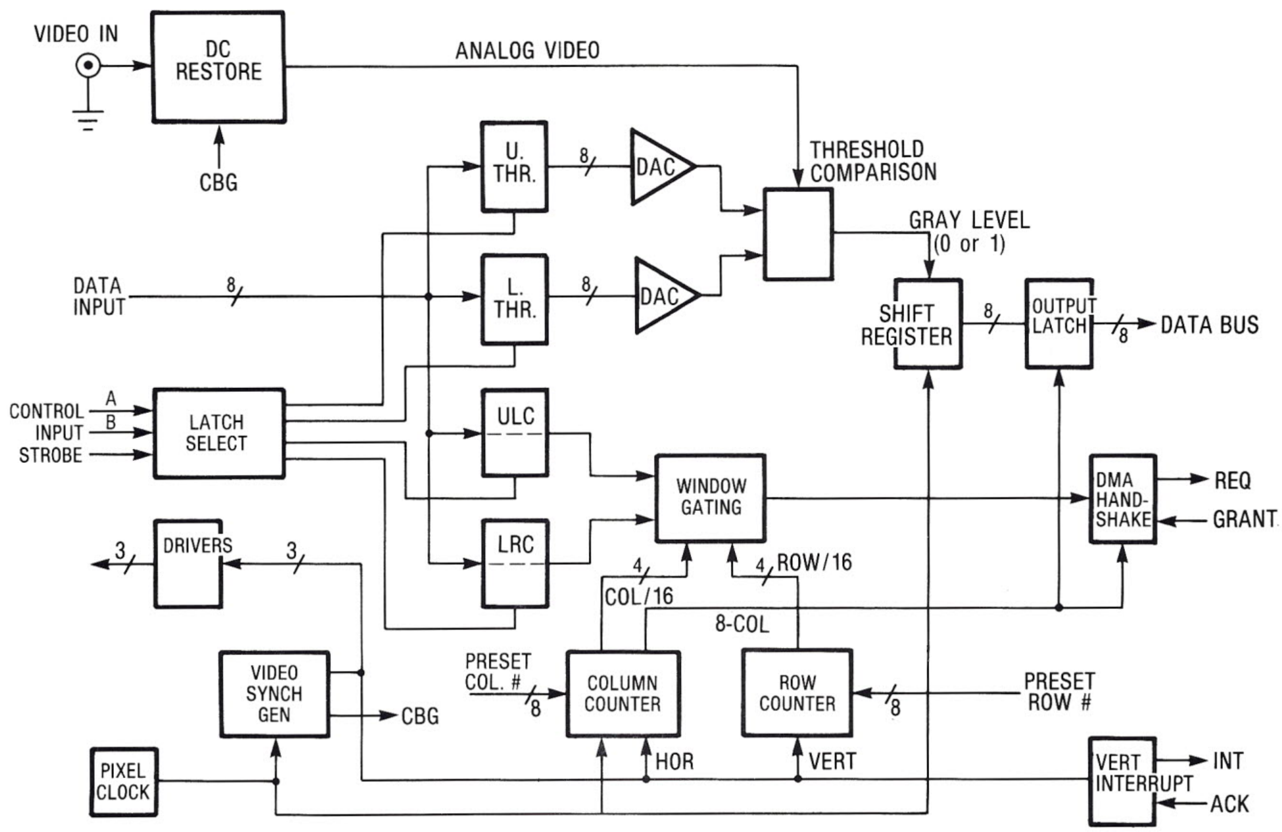


Figure 3. Block diagram of the dual-threshold digitizer. This design presumes that a DMA interface to the computer will be used. Two control bits (A and B) are required to select threshold or window registers for loading. Standard DMA and interrupt handshaking are provided.

latched into 8 bit upper and lower threshold registers. Two additional 8 bit latches hold the two window corner coordinates. The choice of which of the four registers is loaded by the CPU is made by a 1-of-4 selector circuit that requires a two bit control word. The number present on the two control lines when the data strobe from the bus occurs is used to select the appropriate register.

Each threshold register has at its output an 8 bit digital to analog converter (DAC), whose output is scaled so that the digital threshold range of 0 to 255 results in an analog output of 0 to .7V. Since these converters need only respond to changes in the threshold level set by the computer, inexpensive slow converters can be used. The dual threshold comparator outputs a pixel value of "1" only if the normalized video signal level lies between the two thresholds, and zero otherwise. The results of the comparison are clocked into the output shift register by the pixel clock.

Image windowing is achieved by a digital comparator that ensures the current pixel address lies within the rectangle defined by its upper left corner (ULC) and its lower right corner (LRC). To simplify matters, the row and column numbers used in corner coordinates must be a multiple of 16. Since the highest row or column number is always less than 256, this permits each corner coordinate to be expressed in a single 8 bit byte, the four most significant bits (MSBs) determining the row and the least significant bits (LSBs) the column.

The final function to be performed is the transfer of the converted pixel values to the computer. A signal taken from column counter corresponding to one eighth the pixel frequency and is used to deposit in parallel the contents of the pixel shift register into an output latch once every 8 columns (on a multiple of 8). The same signal is used to latch the result of the window comparator onto the DMA request line of the interface, so that no request will be made unless the pixels come from within the window. The handshaking is completed by the interface granting the request, resetting the latch. The request must be granted and the data read from the output register before the next output byte is loaded into the register 8 columns later. For the 5.040MHz pixel clock used in this circuit, a data byte must be read by the computer every 1.6μs during the window. The digitizer requests an interrupt at the start of each field to call the program to prepare for an image transfer.

Circuit Description

Dealing with each of the functions in turn, let's refer now to the actual circuit design to see how each is realized in hardware. I recommend using low-power Schottky TTL chips (74LS series) throughout the circuit, due to their superior time-power performance. The exception is the

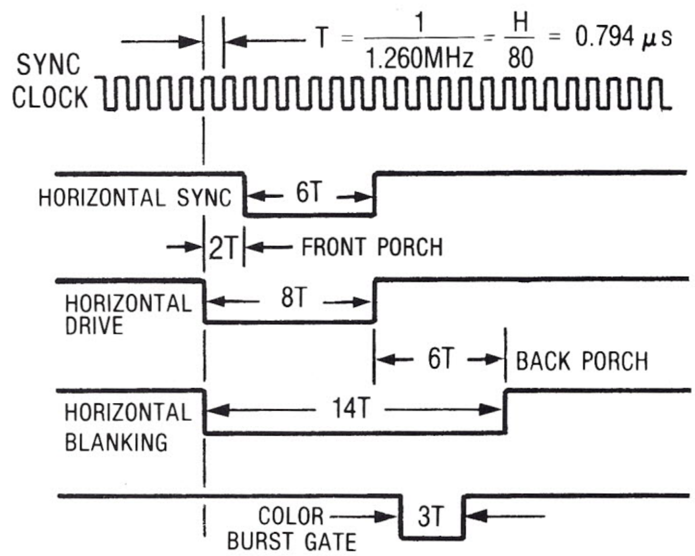


Figure 4. Detail of the horizontal interval. The horizontal drive (HD) signal is provided by the sync generator, lasting 8 sync clock periods. Blanking extends an additional 6 periods (back porch) during which the CBG signal defines the video DC reference.

pixel clock, which is made with 74S04 hex inverters for the sharpest (fastest) clock transitions. The inverter feedback configuration shown makes a reliable clock, but you may also use a standard clock chip or other clock designs, if desired. Although the 74LS (and 74S) devices are preferrable, ordinary TTL will work just as well, but a much stronger 5V supply will be needed if the 74LS chips are not used.

IC11, a 74LS163 synchronous counter, is used to divide the pixel clock by 4 to produce the 1.260MHz sync clock needed by the MM5320N sync generator, IC1. With pin 2 at -12V, the 5320 expects this clock frequency, which is 80 times the EIA standard horizontal rate (horizontal sync every 63.5μs). The relationship between the sync clock and the various signals in the horizontal interval is shown in Figure 4, which gives a precise description of the EIA "front porch" and "back porch" intervals and the CBG interval used for DC restoration.

The back porch, between the rising edge of the horizontal drive and that of the horizontal blanking signal, is of particular significance. When the blanking signal is high, the video signal contains valid intensity information. Since the 5320 only provides composite (horizontal and vertical) blanking, the digitizer uses the horizontal drive (HD) signal and the pixel clock to measure and skip this interval before starting to count the columns of the image. The timing diagram gives the duration of the back porch as 6 ticks of the sync clock, which translates to 24 ticks of our pixel clock. We will use this number later when we choose the preset value of the column counter.

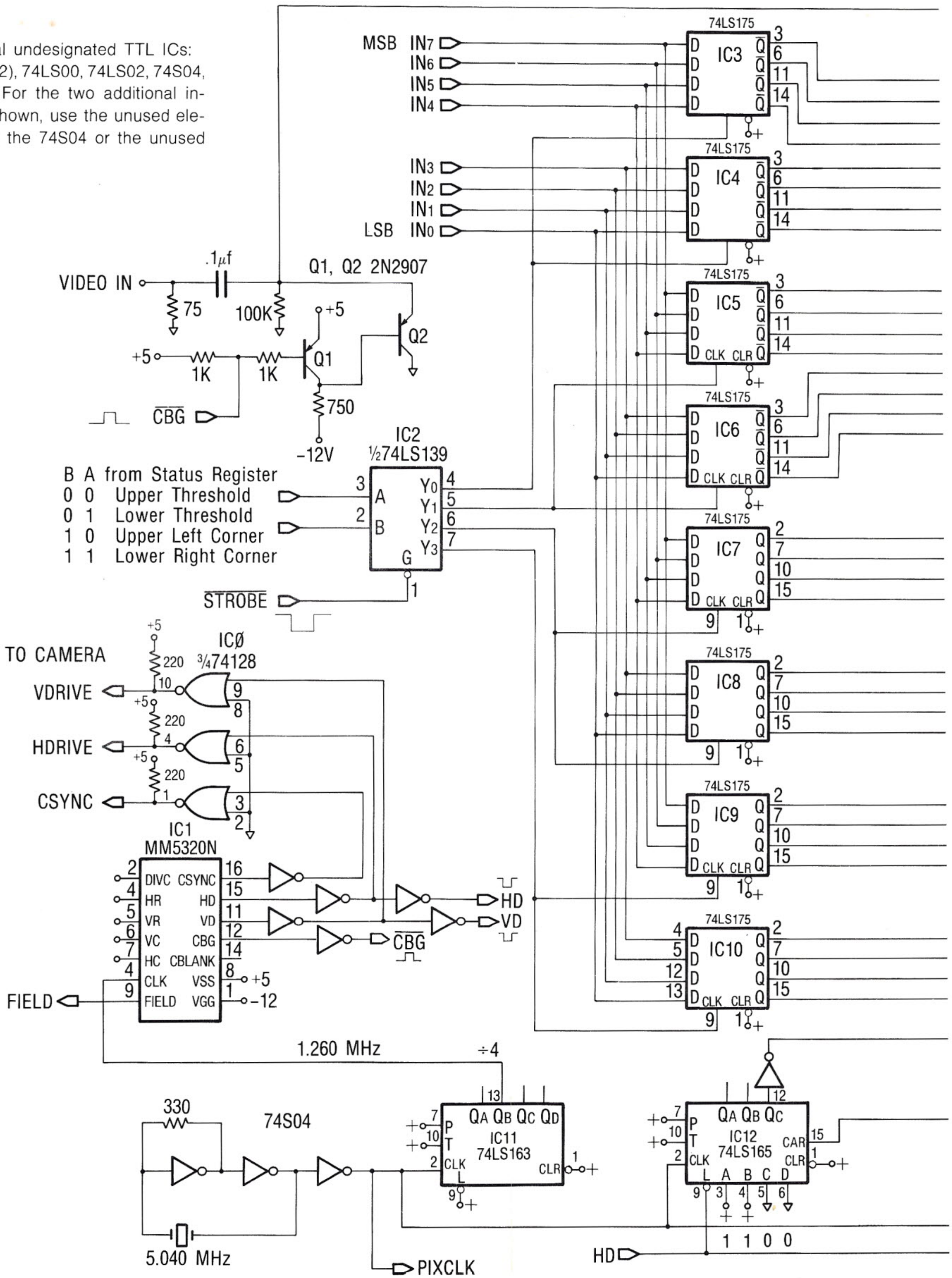
Depending on the type of camera or how you modified it, you will need either composite sync (CSYNC) or both HD and VD signals at the camera. The circuit shows all three of these references converted to signals suitable for 75ohm coax by 74128 drivers (IC0). Another useful piece of information provided by the 5320 is the field index signal on pin 9. This signal goes low for one sync clock period at the start of the vertical drive (VD low) interval, but only on *odd* fields. This information will be made available to the CPU to identify which field a particular window of converted video data came from.

The DC restoration function is realized by transistors Q1 and Q2 (2N2907) and their associated passive network. Although the specs of the 5320 label the signal defining the reference interval as CBG and not $\overline{\text{CBG}}$, the signal is actually an active low. Therefore, an inverter is needed to produce an active high pulse to trigger the DC restoration circuit. When this "$\overline{\text{CBG}}$" is high, Q2 is turned on, clamping its emitter to (near) ground. Once it turns off, the time constant of the input capacitor and the 100K resistor is sufficiently long to hold the DC reference esentially constant until the next update.

The four 8 bit data registers are made using eight 74LS175 4 bit latches, IC3-10, assigned in pairs to the respective functions. Input data lines from the CPU bus are mapped appropriately to their inputs. (The input pinout shown on IC10 is used on all.) The clock (load) inputs of a given pair of latches are connected to one of the four outputs of a 74LS139 selector, IC2, which pulls one output low when strobed by the CPU, loading the selected register. Depending upon your choice of DMA interface, the two input bits that control the selector may be available as control or status bits from the interface, otherwise two bits from some other parallel port must be used.

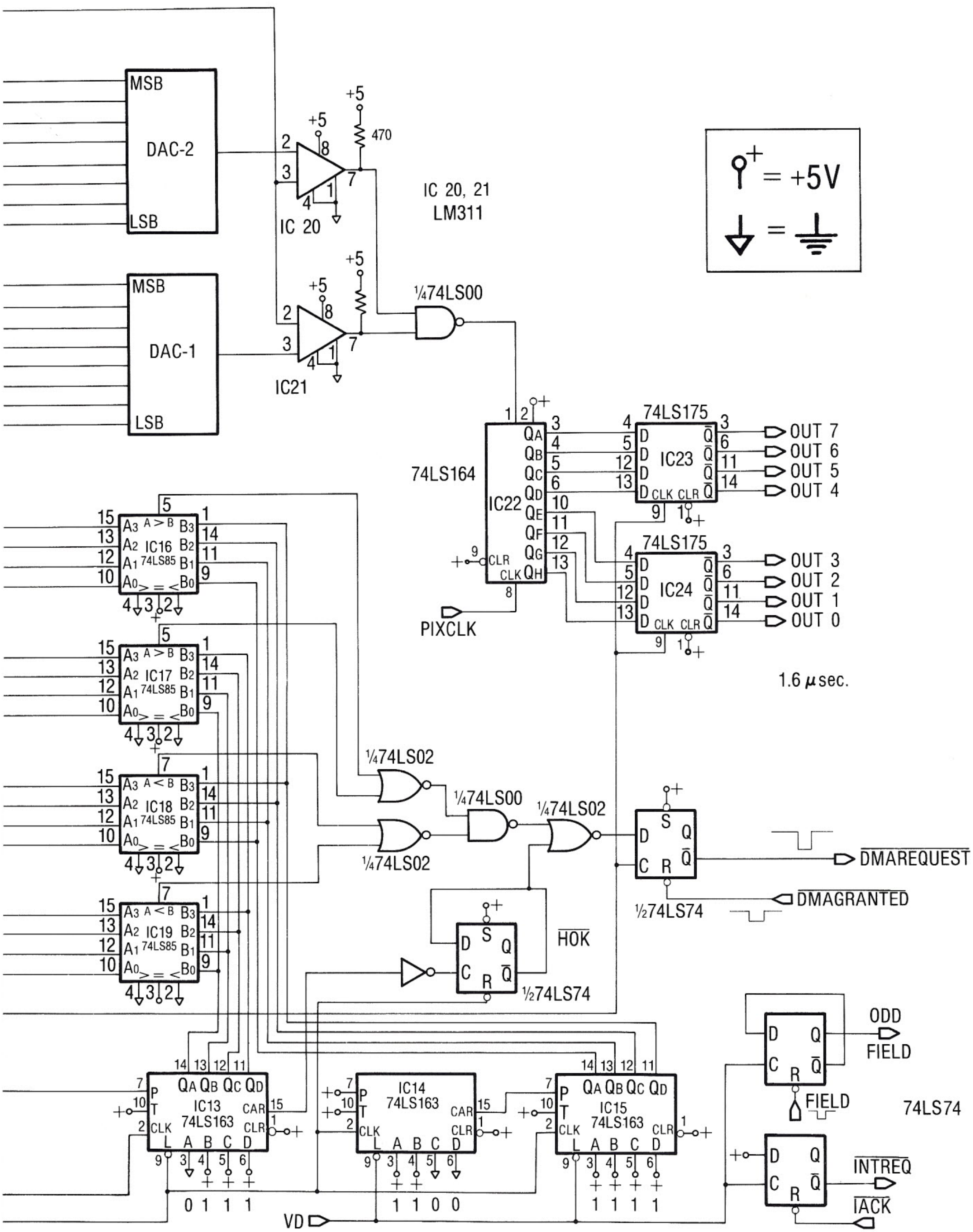
There are a variety of choices available for the DACs used in the digitizer. Since the threshold values held in the latches should be changed no more than once per frame, practically any 8 bit DAC can be used. To keep the cost low, I selected the DAC-IC8BC by Datel (about $5). Two identical converters, configured as shown in Figure 5, are used as DACs 1 and 2 in the digitizer circuit. This converter provides an output current *sink* on pin 4 proportional to the digital input times an externally supplied referece current. This current is translated to a voltage ranging from 0 to .7V by the resistor network shown, but the inverting nature of the current output requires that the *complement* of the digital threshold value held by the latches be used as the input to the DAC. As the value in the latch varies from 0 to 255, the DAC input varies from 255 to 0, so that the translated voltage from the resistor divider varies from 0 to .7V, the standard range of normalized video. If the converter configuration you use

Additional undesignated TTL ICs: 74LS74 (2), 74LS00, 74LS02, 74S04, 74LS04. For the two additional inverters shown, use the unused elements in the 74S04 or the unused gates.



255 − 28 = 227 = (1110 0011)2

COLUMN COUNTER



255 − 12 = 243 = $(1111\ 0011)_2$

LINE COUNTER

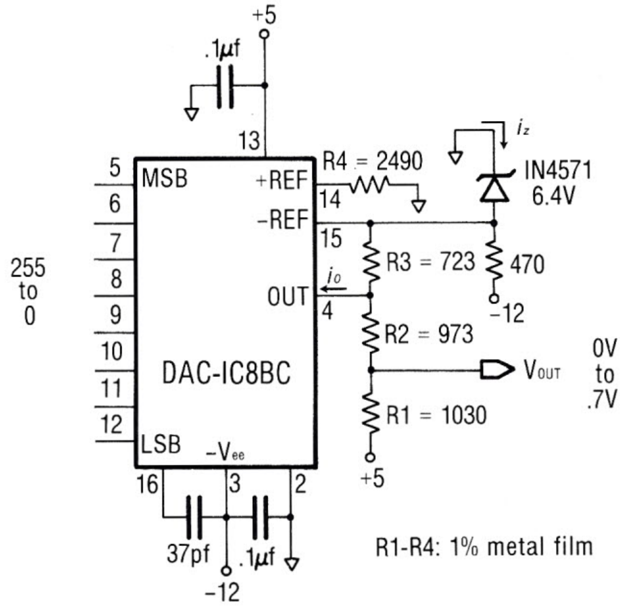


Figure 5. Configuration of the DAC-IC8BC digital to analog converter. Digital input varying from 255 to 0 results in analog output from 0 to .7V. Metal film resistors with 1% tolerance are recommended. Other 8 bit DAC's may be used.

needs the normal binary number (0-255) as input, then use the "Q" outputs of the latches, whose pin numbers are shown on the window latches, IC7-10.

If your robot vision application depends on the absolute accuracy of the thresholds, then the values of resistors R1-R4 in the DAC circuit are critical. R4 and the zener diode establish the reference current for the DAC, whose output is proportional to their values. R1-R3 form the divider that scales the output voltge to video levels. I recommend using metal film resistors with 1% tolerance, both for accuracy and temperature stability. If threshold accuracy is not critical, then carbon resistors may be used.

The output of each DAC is an analog threshold level, which is used as input to a common LM311 comparator (National Semiconductor), ICs 20 and 21, respectively. The output of DAC-1 is on the non-inverting input of IC20, so that the comparator's output is high only if the video level is *below* the threshold. Conversely, the lower threshold level from DAC-2 is on the inverting input of IC21, whose output is therefore high only when the video is *above* the threshold. A NAND gate is used to conjoin the two outputs, so that the input to the pixel shift register, IC22, is the *complement* of the actual pixel value. This is remedied later by using the complement outputs of the output register. The last 8 pixel values clocked serially into the shift register are simultaneously available in parallel on its output pins.

Counters IC12 and IC13 hold the LSBs and MSBs, respectively, of the pixel column number, which is determined by counting the pixel clock. The horizontal drive signal HD is used to preset the counter at the beginning of each new video line. As mentioned earlier, the actual image information does not start at the end of the HD interval, but rather at the end of the blanking interval, 24 pixel periods after HD goes high. Also, note that since the sync clock used by the LM5320 is 80 times the HD rate, there are actually 320 pixel periods between horizontal sync pulses. As shown in the 5320 timing diagram (Figure 4), the horizontal blanking interval is 14T = 56 pixels, leaving 264 pixels of actual image, 8 more than we can use. By discarding the first 28 pixel counts following the HD interval, we skip the back porch and center our 256 columns in the valid video data. This is accomplished by presetting the column counter to 255 − 28 = 227. The period between the two overflows of the column counter (following HD high) define our valid column numbers.

The appropriate preset value for the line counters, IC14-15, is determined similarly. The vertical blanking period lasts 12 horizontal periods longer than the end of the vertical drive (VD low) interval. Since the line count is obtained by counting HD pulses directly, the VD signal is used to preset the line count to 255 − 12 = 243. Since there are only 240 image lines in a field, however, you should restrict your vertical windowing range to line numbers numbered from 0 to 239, inclusive.

The comparison of the 4 bit "nibbles" that define the MSBs of the corner coordinates is performed by four 74LS85 4 bit comparators, IC16-19. The column number MSBs are bussed to ICs 17 and 19, which compare them with the left and right column coordinates, respectively, and the row MSBs go to ICs 16 and 18, for the top and bottom row comparison. To get the windows to range over all the available rows and columns, the comparisons must be *inclusive*. Since the LSBs of the counters don't matter, a zero in a latch must cover a counter range of 0-15, and a latch value of 14 must cover a counter range from 14 × 16 = 224 to 239, etc. This inclusive windowing can be obtained with the comparator outputs gated as shown. The output of the NAND gate is high when any window extremum is exceeded and low otherwise.

The carry output of the column counter (suitably inverted) is used to to toggle a flip-flop (FF) whenever the column number rolls over from 255 to 0. Resetting the FF with HD (which presets the counter) guarantees that the first rollover following the HD interval will define the start of valid image column numbers. A signal to this effect ($\overline{\text{HOK}}$) is gated with the result of the window comparison. The output, high whenever valid video data lies within the desired window, is used to enable a DMA request to the interface. This request, however, must be initiated only

The results of a bi-threshold digitization. White pixels in this photo had grey levels between the two thresholds in the original scene. The use of binary images for robot vision usually requires specially lighted high contrast scenes.

after 8 pixel values have been buffered. Inverting the "4" bit of the column yields a signal that goes from low to high every 8 pixels on a multiple of 8—exactly the moment to latch the contents of the shift register into the output latch and initiate the DMA request.

The digitizer is completed with the addition of one more 74LS74. Both its FFs are clocked by the vertical drive signal VD. One is used to post an interrupt request at the beginning of each field, and the other is toggled by VD to identify the odd field. The field index signal from the LM5320 resets this latter FF to synchronize the odd field signal. These FF's, as well as the output data latches and DMA request FF, may be unnecessary if your DMA interface provides latches that can be used for these functions.

Some other digitizer design options are worth mentioning, too. If you're willing to spend more cash to save some assembly time, you can use a DAC that has data latches built in, such as Datel's DAC-UP8BC (about $14 each), and eliminate latches IC3-6. Refer to the Datel reference manual for details. Another option is to not use dual thresholding at all. If you left out DAC-2 and IC21, connecting the output of IC20 directly to the shift register, then the resulting image in memory will have "1"s for pixels *above* threshold 1 and zero for those below it. If you have access to separate horizontal and vertical sync signals, from a sync separator or other external source, you may use these to synchronize the LM5320 to external sync. (You will also need to preset the sync clock, IC11, appropriately.) Refer to the 5320 data sheet for further information.

The Digitizer in Action

To use the digitizer, it is first necesary to load the threshold and window registers. To load the threshold registers, first set the control bits to select the register you want, then output the desired threshold value to the memory or I/O address assigned to the digitizer interface. The window corner registers are not as simple, since both the row and column numbers of a corner must be encoded into a single 8 bit data byte for output to the digitizer. (If you have an additional control bit available to the digitizer, you may consider altering the design to select the 4 bit latches for the corner coordinates individually.)

Since only the four MSBs of the pixel row and column address are used for window comparison, the row and column coordinates of the ULC and LRC must both be multiples of 16. The *grid coordinates* (m, n) of a corner are the two 4 bit numbers obtained by dividing a corner's image coordinates (row, column) by 16. Since the LSBs are not compared, grid coordinates (m, n) refer to all rows and columns between (16m, 16n) and (16m+15, 16n+15), inclusive. Thus, if the ULC and LRC have identical grid coordinates (m, n), a minimum 16 × 16 pixel window starting at (16m, 16n) will be transferred to the computer. To load a window corner register, place the grid coordinates (m, n) into the left and right (MSB/LSB) halves, respectively, of a single byte, select the appropriate register, and output the byte. Do the same for the other corner. Remember that since the last line of valid image information is on row 239, the highest meaningful row grid coordinate for the LRC is 14, since $14 \times 16 + 15 = 239$ (conveniently!).

Conclusion

Robot vision was once possible only on large, expensive computers, using either slow sampled conversion strategies such as point or column digitizers, or expensive frame grabbers and video image memories. Now, inexpensive digitizers, such as the one just described, are making vision systems affordable to many more schools, groups, and individuals. For example, the Dithertizer II™ by Computer Station is a complete, assembled digtizer with an Apple interface, employing some of the same principles as the one described here, for about $300—within reach of all but the most constrained budgets. Whether you build

(continued on page 19)

Introduction

Anyone who works with images knows how they can burden a small system. An image represented as a 512 × 512 array of picture elements (*pixels*), with 8 bits of brightness (*gray-scale*) information, takes more than a quarter megabyte of storage. In robot vision systems, image understanding often needs multiple images. The system can spend most of its time just shuffling pictures between main memory and secondary storage devices. Clearly, a vision system needs to use either a very large memory or an image with less data.

Even if main memory were large and inexpensive enough to solve the storage problem, raw vision data would still offer a serious drawback: it is not a very convenient form for image understanding. Most image understanding strategies—including those of the human brain—work in a series of successive stages. Each stage performs progressively intricate operations on a progressively smaller amount of more highly structured data. The problem, then, is not only to compress image data. The data must be represented in a form useful to techniques of image understanding.

Can a raw image be represented by only a portion of its data? To answer this, suppose we partitioned an image into regions, each one having just a single brightness level. We could then represent each region by its outline. To reconstruct the whole image we simply fill each outline with the appropriate brightness. Because we can usually describe outlines with much less data than regions, outlines are useful for data compression. Since we can reconstruct the whole region from its outline, outlines can be a good form for data representation.

Chain-code, the subject of this article, is a data structure for storing outlines in computer memory. Chain-code is a list. Its first entries are the x, y coordinates of the point from which to begin tracing the outline. Subsequent entries are numbers giving the direction between each point in the outline and its neighbor. There are eight possible directions between a point and its neighbor. As figure 1 shows, these eight directions are numbered "0" through "7," counter-clockwise. Figure 2 shows an outline and its chain-code.

Edge Detection

Edge detection, for this discussion, refers to the process that takes a standard gray-level image array as input and outputs an array which, ideally, contains only the outlines

of objects in the image.

Though edge detection is still a vital research problem, vision system designers can choose from quite a few techniques currently in use. Rather than discuss these techniques in general, let's follow one particular edge detection process, from start to finish. For those readers who want a more thorough and general discussion of edge detection, I recommend Davis's survey* (*see references*).

Smoothing is the first step in our edge detection process. Smoothing eliminates "salt-and-pepper" type noise. That is, it removes isolated pixels whose gray-level is radically different from those of its neighbors. Noise of this kind can come from defective photosensors or analog-to-digital roundoff error, among other sources. If not eliminated, salt-and-pepper noise can cause spurious edges to appear later, when we compute edge information.

In the next step, the edge detector calculates the brightness difference between each point and its neighborhood. The size of the neighborhood or *window* examined by the edge detector determines how sensitive the edge detection is to noise and to detecting real edges in the image. There is always a tradeoff between the reliability of an edge detector and its speed. For example, a Sobel-type operator, which examines a 3 × 3 window, is shown in figure 3. Though relatively unsophisticated, a Sobel operator has the advantage of being easy to implement in hardware for high-speed edge detection.

When the edge detector has finished determining the brightness differences at every pixel, the image is in the form of a *gradient edge map*. Each pixel in the gradient edge map represents how sharply the gray-level changes at that point.

The next step in our process separates edge points from the rest of the image. The brute force approach to this problem would be to set a single threshold value: Every point with an edge intensity above this threshold would be set to "1," and the others set to "0." But his approach ignores variations in contrast over the image. In regions of high contrast, it would detect spurious edges. In regions of low contrast, it would miss edges. Instead, then, of setting just one threshold for the image, we use a function that sets a *local threshold* for each neighborhood. When the

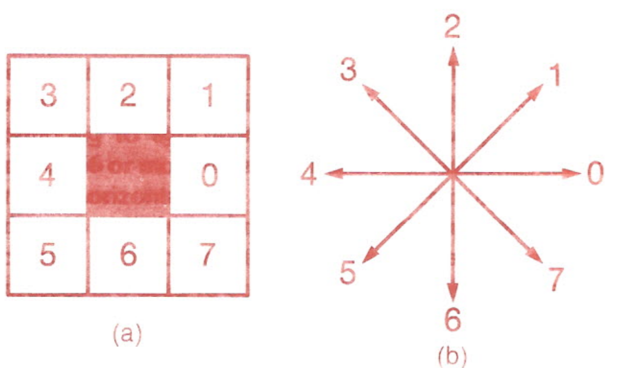


Figure 1. The dark square at the center of (a) represents the pixel we are currently examining. As shown above, the pixel has eight neighbors, numbered "0" through "7" counter-clockwise. As (b) indicates, these numbers can also represent the directions we can travel from the center pixel to its neighbors. Notice that our choice of eight pixels *is somewhat arbitrary. We could construct a four pixel neighborhood—by considering just the four pixels which share a side with the center (numbered "0", "2", "4", and "6"). This would give* four-direction chain-code, *as opposed to the* eight-direction chain-code *discussed in this article.*

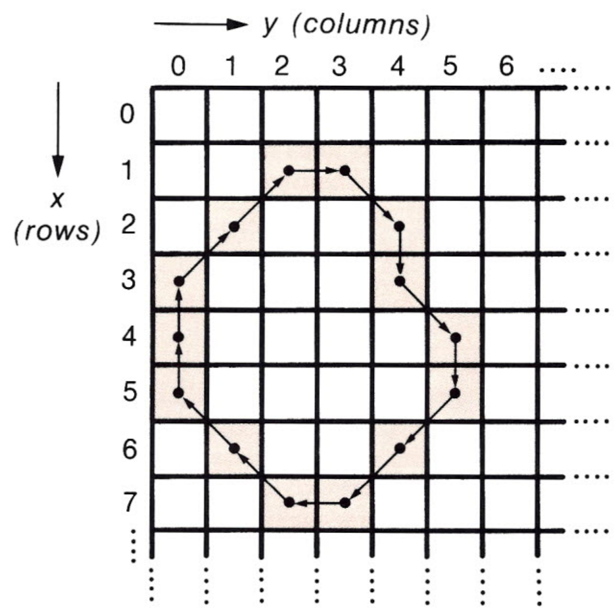


Figure 2. If we use pixel (1, 2) as the first point in the chain, and move clockwise along the curve, we obtain "0, 7, 6, 7, 6, 5, 5, 4, 3, 3, 2, 2, 1, 1" as the chain-code for this outline. We can traverse the chain in the opposite direction by reversing the order of the chain and adding "4" (modulo 8) to each link. "5, 5, 6, 6, 7, 7, 0, 1, 1, 2, 3, 2, 3, 4" is the counter-clockwise chain-code for this curve. Notice that (1, 2) still represents the starting point.

**Editor's Note: In an upcoming issue,* Robotics Age *plans to present some powerful techniques developed since Davis's paper.*

local thresholding has been applied to every pixel, the output image is a *binary edge map*. Every edge point has a value of "1"; every non-edge point has a value of "0". Figure 4 depicts the entire transition of an image—from gray-level array to binary edge map.

Ideally, every edge in the binary edge map would be a simple, closed curve, corresponding to the outline of an object in the image. Unfortunately, edge detectors usually pick up edges that are not on the boundary, and they miss edges that are. To remove spurious edges or complete broken contours, edge detection needs feedback from a process that "knows" how complete outlines in the image should look.

Chain-Encoding

Assume, now, that the image is a binary edge map, as shown in figure 4C. Each pixel in the image array has a value of "1" if it lies on an edge, and "0" otherwise. Notice that the majority of pixels have a value of "0". Therefore, we waste memory when we store the whole array. More importantly, this arrangement wastes processing time because the entire image must be scanned in operations

A	B	C
D	E	F
G	H	I

(a)

(i–1, j–1)	(i–1, j)	(i–1, j+1)
(i, j–1)	(i, j)	(i, j+1)
(i+1, j–1)	(i+1, j)	(i+1, j+1)

(b)

Figure 3. For convenience, we can label a pixel's 3=3 window as shown in (a) or (b). "E" is the pixel under examination, and its neighbors are labelled A, B, C, D, F, G, H, and I, using the labelling shown in (a). Then the Gradient Edge Value, $g(E) = |A + 2B + C - G\ 2H - I| + |A + 2D + G - C - 2F - I|$. The same 3 + 3 window can be used to obtain a local threshold value. Using the labelling shown in (b), let

$$S(i, j) = \sum_{k=-1}^{1} \sum_{L=-1}^{1} W_{kL} * g(i+k, j+L)$$

where the W_{kL} are programmable integer "weights", and W_{00} is minus the sum of the other weights. Each point in the binary edge map is set to one (edge) if $S(i, j)$ is positive, and to zero (no edge) otherwise.

(a)

(b)

(c)

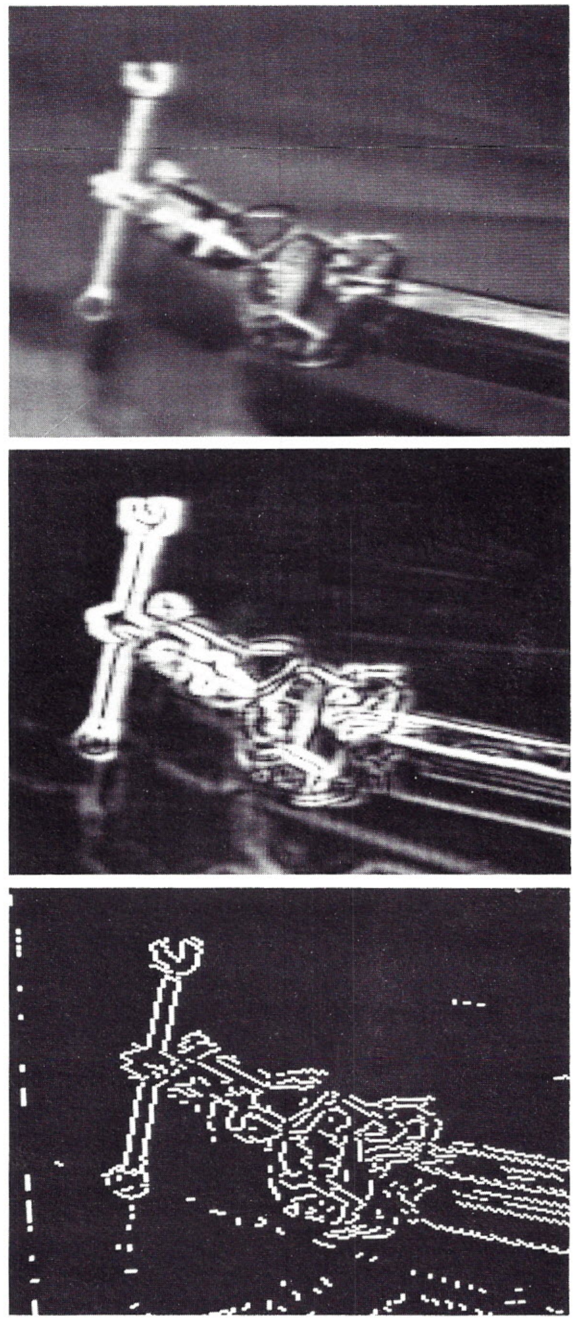


Figure 4. The unprocessed video image in (a) has a resolution of 244 + 188 pixels, with a gray-scale resolution of 256. Image (b) shows the gradient edge map made from image (a). Notice that bright areas in the raw image, have become areas with bright fringes and dark insides in the gradient edge map. Image (c) shows the binary edge map made from image (b). Notice that most edges form broken edge segments rather than simple, closed curves.

which involve only edges. It's time we converted the image to chain-code.

A binary edge map can be chain-coded in three-steps. In the first step, we scan the image for points from which to start a chain. In the second step, we chain-encode all edges that do not form simple, closed loops. The third step then picks up the simple, closed boundaries. Let's consider each step in detail.

In the first step, do the following: Raster-scan the image. At every edge point, count how many of its eight neighbors lie on an edge. A point with no neighbors on an edge is *isolated*. Store it as complete chain-code, since that point is the only one on the edge. A point with one neighbor on an edge terminates that edge. Store it in a list of *end points*. If a point has two neighbors on an edge, it lies in the middle of the edge. Ignore it. Finally, with three or more neighbors on an edge, the point lies at the fork between two edge branches. Store it in a list of *node points*. Figure 5 illustrates this process.

In the second step, we construct the actual chain-code. Do the following: Start each chain with an end-point or a node-point. Examine this point's eight neighbors in clockwise fashion, starting with the immediate righthand neighbor (direction "0"). When a neighbor is found which lies on the edge, add it to the chain. The new edge-point's neighbors are then examined, counter-clockwise, starting in the previous edge-point's direction. In this way, "walk-around" the outline, adding points to the chain. As each point is added, delete it from the image so it will not be picked up twice. Do not delete node-points, though, since more than one edge emanates from them. Continue scanning the neighbors of a node-point, tracing all the edges, until the node-point has no neighbors left in the image. *Then* delete it.

Elegant mathematical relationships exist between an outline and the region it bounds. These enable us to gather certain statistics about a region while walking around its boundary. At every point, we can calculate partial sums for the length of the chain, its "chain-height" and "chain-width", and a set of shape statistics known as *moments*.

For any region, we can generate an infinite series of moments. But for most purposes, the first six terms of the series suffice. The first term of the moment series is the region's area; and the second and third terms are the moments of inertia (familiar from college physics) around the x-axis and y-axis, respectively. The next section discusses how to use moments to recognize objects in the scene. For now, mathematically inclined readers can find out how to calculate moments from outlines in the "Mathematician's Corner" (at the end of this article).

By the third step, the image will be empty of everything but simple closed curves (since the first step ignored points in the middle of an edge). Raster-scan the image until reaching an edge point. Beginning with this point, chain-encode the curve, deleting points as they are added to the chain. By the time we have completely scanned the image, it will be empty, and chain-code for all its edges will reside in memory.

Image Understanding

By now, the vision system has converted the image to a collection of chain-coded edges and some statistics about them. The question remains: How can this information be used to recognize objects?

One commonly used recognition technique is *template-matching*. As the name implies, template-matching compares the outline of an object in the image with stored outlines ("templates") of known objects.

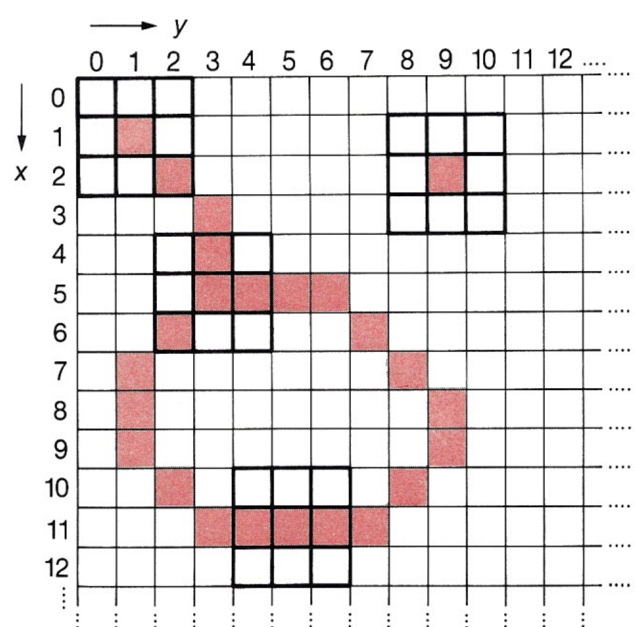


Figure 5. The 3 × 3 window grids highlight the four types of edge points in an image. The point with coordinates (2,9) is an isolated point. None of its neighbors lie on an edge. It can be considered a complete edge in itself. (1, 1) is an end point. With only one neighbor "turned on", it terminates the edge. (5, 3) is a node-point, since more than two of its neighbors are "on". Notice how the node-point forms a fork between two edge segments. (11, 5) has two "turned on" neighbors. It therefore lies in the middle of an edge.

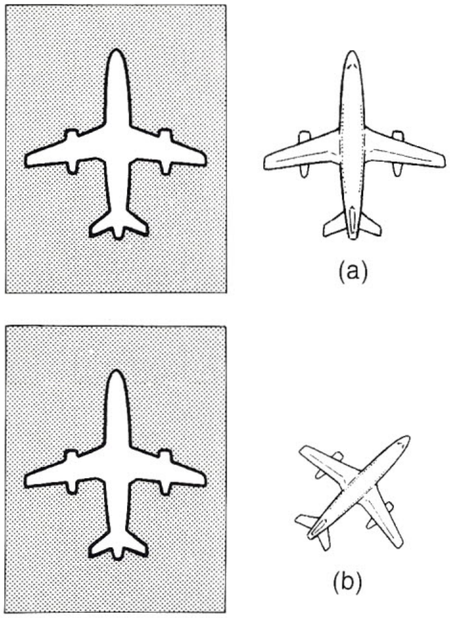


Figure 6. The airplane on the right side of (a) can be recognized by matching it with its template (on the left side), stored in computer memory. The airplane in (b), however, must be rotated and rescaled if it is to be compared with the template lying to the left of it.

Figure 6, which illustrates this process, also reveals one of its weaknesses: Before an object can be compared with a template, it has to be moved, rotated, and rescaled to fit the template's frame of reference. To prepare an outline for template-matching, we need to define coordinates which depend only on the outline itself; not on the outline's location in the image. Making the outline's center-of-mass the origin of its coordinate system is the first step. The next is to orient the outline along an axis intrinsic to the outline, not its orientation in the image. The *axis of minimum moment of inertia* is one such axis, which we can calculate from moments. (Remember, we calculated the first six moments as we chain-encoded the outline). Finally, we can scale the outline by adjusting its "chain-height" to the template's height. The reader can find formulas for "normalizing" chain-coded outlines in the "Mathematician's Corner".

Discrimination is another, more sophisticated method of object recognition. It also requires an internal model of the object being sought. However, this model is a list of "features". Average brightness, center-of-mass, region area, and any other measurable quantity associated with an object may be included in the feature list (often called a *feature vector*). The most useful features, though, are those which remain constant no matter how an object is moved around or rotated in an image.

Moment invariants—certain linear combinations of moments—are features which have this trait. No matter how an object is translated, rotated, or rescaled in an image, moment invariants remain constant (though digitization and approximation errors do creep in). Like the moment sequence itself, an infinite number of moment invariants can be generated. We may need only a few, however, for a given recognition task. The curious reader can turn to the "Mathematician's Corner" to see what the first few moment invariants look like.

Moment invariants can characterize objects in two-dimensional scenes—where we can ignore depth information. Wong and Hall (*see references*) reported that moment invariants helped classify printed characters with an accuracy of 95%. They also mention that moment invariants have been used to characterize X-rays and identify aircraft. In their own work, Wong and Hall used moment invariants to match aerial radar images with optical images of the same scene.

Limitations

Chain-code has many advantages: It is compact, easily constructed, easy to understand, convenient for some kinds of mathematical manipulation, and useful for some types of statistical object recognition. It has, though, several important limitations.

One problem is that we are unable to construct chain-code in a single raster-scan of the binary edge map. This is a drawback in a robot vision system which must analyze changing scenes in real-time. A number of researchers are trying to speed up the chain-encoding process.

Cederburg (*see references*), however, has proposed discarding chain-code in favor of a different, chain-code-like structure called RC-Code (for Raster Chain-Code). RC-Code uses only the four chain-code directions that lie in the "forward scan" direction (the chain-code directions 0, 7, 6, and 5). RC-Code has the advantage of being constructable in a single raster-scan of the edge map.

Certain basic operations, such as arbitrary rotations and scale changes, are very awkward to perform on chain-code. Rotations which are multiples of 90 degrees are simple enough: just add (in module 8) multiples of 2 to each link in the chain. For all other rotations, however, we must transform the chain-code back into coordinates, rotate the coordinates, and re-encode the chain. Most scale changes also force us to convert back to coordinates.

Chain-code stores outlines; but outlines are not the only way to represent an image. Sometimes, we might prefer to work with two-dimensional regions in an image. When we want to store a region in computer memory, we can use a data structure that records the sizes and number of two-

The Mathematician's Corner

Notation

Let x and y be variables, and let (x, y) be the coordinates of an arbitrary point (pixel) in the image. Let x_L and y_L represent the value of x and y, respectively, at link L along the chain-coded curve. (x_0, y_0) is taken to be the starting point for the curve.
For any $L \geq 1$, let

$$\Delta x_L = x_L - x_{L-1} \qquad \Delta y_L = y_L - y_{L-1}$$

If a curve is approximated by an arbitrary polygon, Δx_L and Δy_L can take on any value. By adopting the notation given here, we make most of the formulas given below apply easily to any polygonal approximation of a curve. Chain-code can be viewed as a special case of polygonal approximation, where Δx_L and Δy_L can take only the values -1, 0, or 1.

The Length of a Chain

An even link is defined as a link whose chain-code is 0, 2, 4, or 6. An odd link, conversely, is one whose chain-code is 1, 3, 5, or 7.
Let n_e be the number of even links in a chain, and n_o be the number of odd links. Then, the length of the chain is

$$\text{LENGTH} = n_e + \sqrt{2} * n_o$$

Chain-Height and Chain-Width

Let

$$X_i = \sum_{L=1}^{i} \Delta x_L + x_0 \qquad Y_i = \sum_{L=1}^{i} \Delta y_L + y_0$$

Then,

$$\text{CHAIN-WIDTH} = \max_i X_i - \min_i X_i$$

$$\text{CHAIN-HEIGHT} = \max_i Y_i - \min_i Y_i$$

If a box were drawn around the curve so that the curve touched all four sides, then the width of the box would be the chain-width, and its height would be the chain-height.

Moments Defined for a Region

In general, a $(p + q)$th order moment, m_{pq}, is defined by the equation

$$m_{pq} = \iint_{-\infty}^{\infty} f(x, y)\, x^p\, y^q\, dxdy,$$

where $f(x, y)$ is a density distribution function. In the case of a uniformly bright region in an image, we can take $f(x, y)$ to be 1, if (x, y) lies in the region and 0, otherwise. For a uniformly bright region, **R**, the definition of its $(p + q)$th order moment becomes:

$$m_{pq} = m_{pq}(\mathbf{R}) = \iint_{\mathbf{R}} x^p y^q dxdy.$$

Center of Mass and Angle of Minimum Moment of Inertia

We can use moments to compute the centroid, $(\overline{x}, \overline{y})$ of a region by

$$\overline{x} = \frac{m_{10}}{m_{00}}, \quad \overline{y} = \frac{m_{01}}{m_{00}}$$

We can also use moments to compute the angle of a region's axis of minimum moment of inertia, Θ. This quantity defines a region's orientation within a two-fold degeneracy. It is given by

$$\Theta = \frac{1}{2} \tan^{-1} \left[\frac{2\,(m_{00} m_{11} - m_{10} m_{01})}{(m_{00} m_{20} - m_{10}^2) - (m_{00} m_{02} - m_{01}^2)} \right]$$

The quantities x, y, and Θ are useful in object recognition because they specify the position and orientation of regions. This, in turn, allows us to define a transformation between image and template coordinates.
If a region is symmetrical, its Θ has further degeneracies. In the extreme case, for a perfectly circular region, every axis that passes through its centroid is an axis of minimum moment of inertia. Because of digitization errors (etc.), though, regions are rarely symmetrical.

Calculating Moments

The moments for a region can be calculated while "walking around" its boundary. Let n be the number of links in a chain-coded (or polygon-approximated) curve.

Let

$$A_L = x_L\, \Delta y_L - y_L\, \Delta x_L$$

Then, the first six moments are given by

$$m_{00} = \frac{1}{2} \sum_{L=1}^{n} A_L$$

$$m_{10} = \frac{1}{3}\sum_{L=1}^{n} A_L \left(y_L - \frac{1}{2}\Delta y_L\right)$$

$$m_{01} = \frac{1}{3}\sum_{L=1}^{n} A_L \left(x_L - \frac{1}{2}\Delta x_L\right)$$

$$m_{20} = \frac{1}{4}\sum_{L=1}^{n} A_L \left(x_L^2 - x_L\Delta x_L + \frac{1}{3}\Delta x_L^2\right)$$

$$m_{11} = \frac{1}{4}\sum_{L=1}^{n} A_L \left(x_L y_L - \frac{1}{2}x_L\Delta y_L - \frac{1}{2}y_L\Delta x_L + \frac{1}{3}\Delta x_L \Delta y_L\right)$$

$$m_{02} = \frac{1}{4}\sum_{L=1}^{n} A_L \left(y_L^2 - y_L\Delta y_L + \frac{1}{3}\Delta y_L^2\right)$$

Moment Invariants

Moment invarients are best calculated in several stages. In the first stage, we find quantities called central moments—the region's moments in its center-of-mass coordinate frame. For a uniformly colored region, **R**, its $(p + q)$th order central moment, μpq is defined by

$$\mu_{pq} = \iint_{\mathbf{R}} (x-\bar{x})^p (y-\bar{y})^q \, dxdy$$

The first few central moments are

$$\mu_{00} = m_{00}$$
$$\mu_{10} = 0$$
$$\mu_{01} = 0$$
$$\mu_{20} = m_{20} - \bar{x}m_{10}$$
$$\mu_{02} = m_{02} - \bar{y}m_{01}$$

After we find the low order central moments, we calculate the normalized central moment, ηpq, by the equation.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{s}} \quad \text{where } s = (p + q)/2$$

Finally, we can calculate some low order moment invariants. Let φn be the nth moment invariant. Then

$$\varphi_1 = \eta_{20} - \eta_{02}$$
$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\,\eta_{11}^2$$
$$\varphi_3 = (\eta_{30} - 3\,\eta_{12})^2 + (3\,\eta_{21} + \eta_{03})^2$$
$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

I leave it as an exercise for the reader to prove that these quantities are invariant with respect to translation, rotation, an scale change. Ambitious readers should turn to Hu and Wong and Hall (*see references*) to further explore the mathematics of moments.

dimensional blocks which cover that region.

The most serious limitation of chain-code, though, is that it is not a good structure for representing three-dimensional scenes. It works well enough in two dimensional situations—when aerial scenes are analyzed, for example, or when parts on a conveyor belt are inspected from directly overhead. However, we need other methods when we need to model the three-dimensional world. And the problem of modelling the three-dimensional world—in all its glory—has not yet been conclusively solved. R

References

[1] Cederberg, R. "Chain-Link Coding and Segmentation for Rater Scan Devices," *Computer Graphics and Image Processing*, Vol. 10, 1979, pp. 224-234.

[2] Davis, L. S. "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing*, Vol. 4, 1975.

[3] Eskenazi, R. and Wilf, J. Low Level Processing for Real-Time Image Analysis. Publication 79-79, Jet Propulsion Laboratory, Pasadena, Calif., Sept. 1979. This describes in detail a hardware implementation of real-time edge detection.

[4] Freeman, H. "Computer Processing of Line-Drawing Images," Computing Surveys, Vol. 6, No. 1, March 1974, pp. 57-97. This is the standard reference on chain-code. I have heard it called "the chain-code bible."

[5] Hu, M. "Visual Pattern Recognition by Moment Invariants," IRE Transactions on Information Theory, IT-8, 1962, pp. 179-187. This is a highly mathematical exposition on the theory of moment invariants.

[6] Wilf, J. and Cunningham, R. Computing Arbitrary Moments from Chain-Coded Boundaries. Publication 79-49, Jet Propulsion Laboratory, Pasadena, Calif., Nov. 1, 1979.

[7] Wong, R. and Hall, E. "Scene Matching with Invariant Moments," *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 16-24.

[8] Wong, V. Computational Structures for Extracting Edge Features from Digital Images for Real-Time Control Applications. PhD. Thesis, Division of Engineering, California Institute of Technology, Pasadena, California, Feb. 1979.

Video Input Signal
(cont. from page 11)

the digitizer described here or decide to buy one, one message is clear: microcomputer-based robot vision has arrived! ®

References

[1] *The TTL Data Book*, Texas Instruments Inc., Dallas, Texas. This is the professional design engineer's complete TTL reference.

[2] *TTL Cookbook*, by Don Lancaster. H. W. Sams & Co., Indianapolis, Ind. If you have never built TTL projects before, *this is essential reading*. Although a little dated (1974), it provides an excellent introduction to logic design and proper circuit layout and construction techniques.

[3] *MOS/LSI Databook*, National Semiconductor Corp., Santa Clara, Calif. Contains LM5320-5321 specifications and timing diagrams.

[4] *Engineering Product Handbook*, Datel-Intersil, Inc., Mansfield, Mass. Contains specs and application information for 8 bit DACs DAC-IC8B and DAC-UP8B.

CAMERA GEOMETRY

for Robot Vision

Relating what the camera looks at to what the camera sees.

by
Alan M. Thompson
Robotics Age Magazine

The Image-Object Relationship

Most articles dealing with robot vision are primarily concerned with the problem of detecting, describing, and recognizing target objects in an image, usually the digital representation of an image taken by a TV camera. In this article, we will take a look at the geometric relationships between objects in the real world outside of the robot and their images in the digitized picture. Clearly, knowing these relationships is essential for tasks that require the robot to *physically* interact with its environment, as opposed those where it just *passively* derives information by looking at a scene. Even in the latter case, it is often necessary to describe the location of objects quantitatively.

In general, robots must deal with multiple coordinate systems. Manipulator control programs typically describe the location of the robot's hand, as well as target points, relative to the base of the arm. The location of machine tools or part-feeders, though, may be expressed in terms of an absolute *workspace* frame. For a mobile robot, the navigation program must be concerned with the location of the robot in some universal frame. For each of these problems, the robot designer is faced with providing coordinate transforms that allow location measurements made in one frame to be expressed relative to another.

For a vision-equipped robot, the relationship between the real world and the two-dimensional world of an image presents special problems due to the nature of the imaging process. Not only can distortions of perspective complicate the problem of recognizing objects in the general case, but these distortions also complicate the problems of automatic range measurement and of accurately determining spatial relationships from visual information.

One of the critical pieces of information needed by a vision subsystem is a description of the camera—for only with this knowledge can the location of targets, whether landmarks or workpieces, be derived from their position in the image that the camera sees. Assuming that the robot has identified a target point in an image (by some perceptual or cognitive process such as those I described in an earlier article [1]), the *geometric transformation* between the object world and the image can be used to help compute the target's real-world coordinates. Similarly, if we know the target's coordinates relative to the camera, we can use the transformation to tell us where to find the target in the image. As I will show, some methods of automatic range measurement by binocular vision require a combination of these operations, applying the

transformation in both directions.

The tools needed to derive the image-object relationship come from the broader field of *projective geometry*, which finds practical application in a number of important areas, including mapmaking and computer graphics. It is used here to find the projection that maps points in three-dimensional space into a plane by projecting them through a single point. The 3-space is, of course, the real world; the plane, the focal plane of the camera. The point used for the projection is the camera's *focal center*, defined below, and the resulting mapping is called the *perspective transform*.

The Camera Model

Before we can start writing equations to describe the perspective transform, however, we must first have a mathematically precise description, or *model* of the structure of a camera. To avoid having to deal with the complexities introduced by lenses, we'll start by limiting our description to the simplest type of camera, the pinhole camera. (Figure 1)

In a pinhole camera, the image of a point in 3-space is made by a ray of light that leaves the point, passes through the pinhole, and intersects the rear plane of the camera, presumably leaving an image on the film. For an "ideal" pinhole, the ray of light from the point is an ideal line, so that the image of the space point is itself a point, eliminating the need to focus. The pinhole, of course, is the camera's focal center, which serves as the projection point. Let the vector **C**, measured in some convenient external coordinate system, describe the location of the focal center.

The next thing we do is define the location of the image plane relative to the focal center. In a film camera, this is the location of the film that records the images of all the points in the field of view of the camera. For a video camera, it is the sensitive surface of a vidicon tube or solid-state imaging array. The orientation of the image plane can be described by a direction vector orthogonal to it, shown in the figure as the camera's *aiming vector*, **A**. The *focal distance, f,* measured from the focal center to the image plane along the vector **A**, is an important factor in determining the size of an object's image.

To locate points in the image plane, we must define two coordinate axis vectors **H** and **V**, orthogonal to **A**, that correspond in this case to image horizontal and vertical directions, respectively. For a film camera, the orientation of these axes is rather arbitrary, but for a video camera, the vectors' orientations are determined by the manner in which the image is scanned and digitized, as will be explained.

By using the three direction vectors **A**, **H**, and **V** to define a coordinate system origined at the focal center **C**, our model of the pinhole camera is complete. (Figure 2) We can use our camera-center frame to provide an exact description of the location of the image plane in space: the plane can be defined as

$$\mathbf{C} - f\mathbf{A} + u\mathbf{H} + v\mathbf{V},$$

where u and v are scalar parameters that can be used as the coordinates of points in the image plane. Of course, for

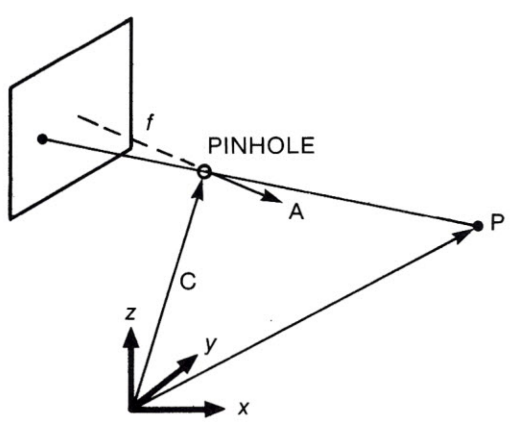


*Figure 1. A pinhole camera. The image of point **P** is formed by a ray of light that passes through the pinhole at **C**. The aiming vector **A**, and the focal distance f, determine the location of the image plane relative to **C**.*

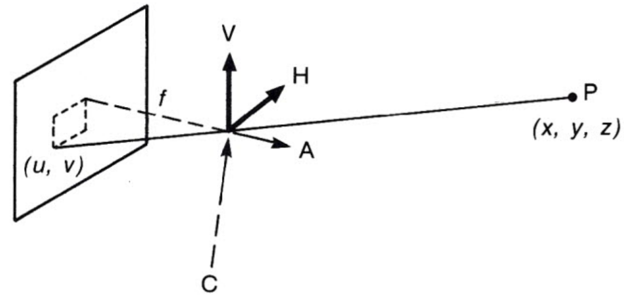


*Figure 2. Image horizontal (**H**) and vertical (**V**) vectors serve as coordinate axes that permit the location of points in the image plane to be measured. Here, the image of point **P** is located at image coordinates (u, v).*

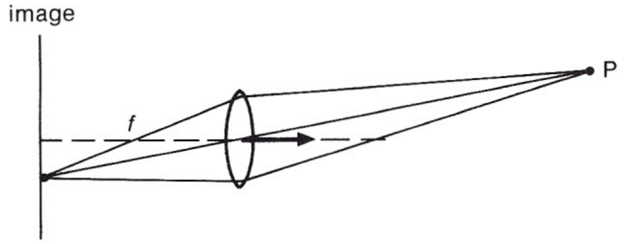


Figure 3. With the addition of a lens, a cone of light from **P** *is focused on the image plane, producing a brighter image. Although the focal distance f needed for focusing varies with the distance to the target, the basic pinhole camera model holds.*

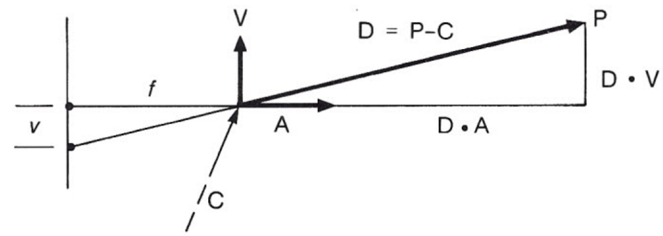


Figure 4. A side view of the imaging process. The components of **D** *along the* **A** *and* **V** *axes are found by taking the scalar product, forming a right triangle as shown. One side of a similar triangle is given by the focal distance, so that v, the image coordinate in the direction of* **V**, *may be computed.*

a real camera the sensitive part of the image plane is bounded in *uv* space, thereby defining the field of view of the camera. With this model we can accurately say that for every point **P** in the field of view, there is a corresponding image of the point located in the image plane at some *image coordinates* (u, v). We are now equipped to examine the geometry in detail to find the actual relation between the target point **P** and its unique location in the image.

Although our camera model was derived by assuming a pinhole camera, it turns out, fortunately, that it also holds quite well for most general purpose lenses, too. Figure 3 shows a side view of a camera with a larger aperture that uses a lens to focus the image of the point **P** onto the image plane. In this case, all the light from **P** that falls on the lens is concentrated on the image point, producing a much brighter image. For fixed-focus cameras, the camera model remains identical, and the focal distance f corresponds to the focal length of the lens. In variable focus cameras, however, the lens can be moved further away from the image plane to focus on points close to the camera, thereby moving the focal center and changing the focal distance. Lenses with multiple elements are complex to describe, and with them, the images of straight lines in space may appear curved (wide-angle, "fish-eye" lenses, etc.). For these lenses, this simple *linear* camera model does not hold.

The Camera's Perspective Transform

Let's start by taking a side view of the camera geometry and examining just the vertical component of the projection. In Figure 4, the camera's aiming vector **A** and image vertical vector **V** lie in the plane of the paper. First, we form the displacement vector **D** from the camera's focal center **C** to the target point **P**. We can *project* this vector onto the **A** and **V** axes by taking the scalar (dot) product of **D** with the respective unit vectors. The resulting scalars give the lengths of two sides of a right triangle as shown. Since **V** is parallel to the image plane, the solution for the vertical component v of the image of the point is obtained by noticing that a similar right triangle is formed by v and the camera's focal distance f. Thus, since the sides of similar triangles are proportionately scaled, we can say that:

$$v = f\,\frac{\mathbf{D}\cdot\mathbf{V}}{\mathbf{D}\cdot\mathbf{A}}\,, \quad \text{where } \mathbf{D} = \mathbf{P} - \mathbf{C}$$

By a similar process, taking a top view of the geometry, substituting the image horizontal vector **H** for **V**, we can solve for the image horizontal coordinate of the image of **P** as:

$$u = f\,\frac{\mathbf{D}\cdot\mathbf{H}}{\mathbf{D}\cdot\mathbf{A}}$$

These two equations are all it takes to find the image of a known target point in the image plane, provided we know the 3-space coordinates of the target and the camera parameters f, **C**, **A**, **H**, and **V** that define the location of the camera center and the image plane in the same 3-space. Determining these parameters requires a process of camera *calibration* that we'll return to later.

Note that, although we assumed that **A**, **H**, and **V** were unit vectors to aid the derivation, the resulting expressions for v and u, and, in fact, the actual derivation, are independent of their respective lengths, as long as their lengths are the same. This fact will become significant later when we discuss the calibration procedure.

So far we haven't said anything about the fact that for robot vision, the image from the camera must be digitized and made available to a computer. The relations we derived for the image coordinates of a target point measure the coordinates in the same units used to measure the camera's focal distance, f. This may be fine for film cameras where the negative itself provides a record of actual image measurements, but for a digitized picture we must consider the additional mapping from image plane to the representation of the image provided by the digitizer.

Describing the Digitized Image

Typically, an image digitizer partitions the image plane

into a rectangular array of picture elements, or *pixels*. The number of bits of information per pixel is not important here; all we are concerned with is the correspondence between the rows and columns in the digitized picture and points on the imaging surface. For a solid-state imaging array, the correspondence between the imaging surface and the row and column is fixed geometrically by the device's pattern of light-integrating cells, providing the most stable mapping. For a vidicon tube, however, the scanning of the image is accomplished by deflection coils, and the mapping is subject to drift. The digitizer counts the number of video lines, and the column number is obtained by counting a "pixel clock" that also triggers the sampling and conversion of the video signal. (*See article in this issue.*) The row and column numbers provided by the digitizer are typically used to index the memory address at which the digitized pixel information is stored, and the computer program keeps track of the correspondence between the stored data and the original pixel address in the image.

The actual mapping from the pixel address to the physical image coordinates is determined by four quantities: the pixel address (row and column)of the element nearest the origin of the uv coordinate frame (i_o, j_o), and the sampling *resolution*, the number of pixels per unit length in the image horizontal and vertical directions, given by n and m, respectively. The mapping can then be written as:

$$u = n(j - j_o) \quad \text{and} \quad v = m(i - i_o).$$

It is highly desirable for n and m to be the same, that is, that the pixels should be "square" instead of "rectangular". If this is the case, then some pattern-matching processes are simplified because a silhouette will appear to have the same proportions when rotated to a different orientation in the image plane. The scan characteristics of a vidicon camera can be altered to achieve this, but not all solid-state cameras have this feature.

It turns out to be extremely simple to combine this additional transformation from image to pixel coordinates into our original mapping from 3-space into the image. Equating the expression for v in terms of the row i with the vertical term of our original transform and solving for i, we get:

$$i = \frac{f\,\mathbf{D{\bullet}V}}{m\,\mathbf{D{\bullet}A}} + i_o, \text{ which can be written as}$$

$$i = \frac{\mathbf{D{\bullet}V'}}{\mathbf{D{\bullet}A}}, \text{ where } \mathbf{V'} = \frac{f}{m}\mathbf{V} + i_o\mathbf{A}$$

By redefining **V** we can still use the simple equation for the projection and completely account for the origin and resolution of the image digitization. A similarly redefined image horizontal vector **H'** is used in an expression for j. For the rest of this discussion, **H** and **V** will refer to these redefined vectors.

It is worth pointing out that due to the slight diagonal slope of the horizontal scan in a vidicon tube, caused by the continuous vertical motion of the beam, the horizontal and vertical axes of the image are not necessarily orthogonal. While this does result in some shape distortion since the pixels aren't exactly square, it doesn't affect our equations for the perspective transform.

Locating Targets in 3-Space

The perspective transform is a valuable tool with many applications in computer graphics and image processing. With it, any three dimensional model of an object or scene can be projected into a plane for graphical display, allowing it to be viewed from different angles. For robot vision, the expressions we have derived so far can be used to tell us where in an image to look for targets when we know their location in some external 3-space frame.

More typically, however, we are concerned with giving the robot the capability to see something in an image and then compute its location in the surrounding environment. Unfortunately, this is not as simple as computing the projection was, because the image only supplies two pieces of information, the row and column coordinates, for any real-world target point whose projection it contains. More specifically, target points lying on the same line through the camera center, but at different ranges from it, will all appear at the same image coordinates. Without additional information, one camera alone cannot determine range.*

Even so, by supplying a somewhat arbitrary constraint we can use the image coordinates to solve for a direction vector that points to the target from the camera center. This pointing vector can then be used for various triangulation schemes. The length of the vector **D** used in the equations for u and v is the range r from **C** to the target. Since we don't know this range, we can make the assumption that $\mathbf{D \bullet A}=1$. This allows one component of **D**,

*Actually, monocular ranging schemes have been devised that rely on restricting the depth-of-field of the camera's focus and then writing software that attempts to decide when the image of the target is in focus. Calibrating the focus control of the camera can then be used to obtain an approximate range measurement.

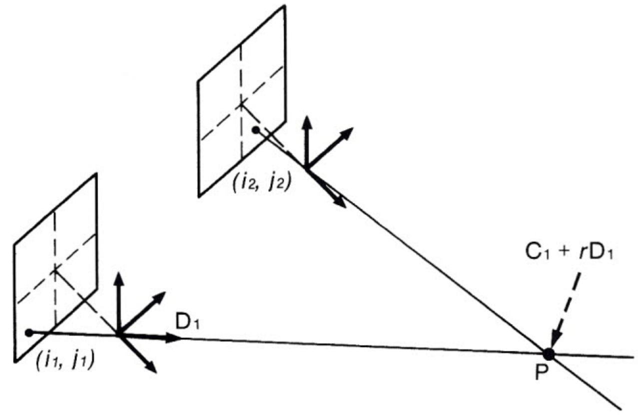


Figure 5. Finding the image of a target at (i_1, j_1) in camera 1 allows a direction vector, $\mathbf{D}_1$, to be computed. Seeing the same target in camera 2 at (i_2, j_2) allows its range from either camera to be determined.

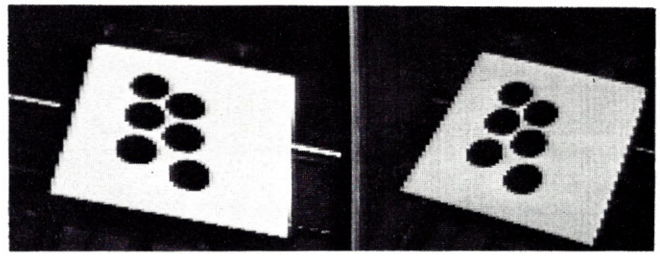


Figure 6. An actual stereo pair taken by adjacent TV cameras. The horizontal shift of a target point, between its coordinates in the left and right images, is the primary range clue.

to be written in terms of **A** and the other two components. Knowing the two image coordinates *i* and *j* of a target point allows us to write two equations for the two remaining components of **D**, which can then be solved algebraically.

If the same target point can be identified in different calibrated cameras, then the (approximate) intersection of lines emanating from the camera centers in the respective directions can be computed to solve for range. Due to various errors (round-off, digitization, etc.) an exact intersection may not exist, so this computation can be cumbersome. A simpler, though less general method, can be used if the two cameras are displaced in the approximate direction of the image horizontal axis.

In Figure 5, two cameras are shown with a lateral separation along **H**. Target point **P** has an image at (i_1, j_1) in camera 1, which is used to solve for direction unit vector $\mathbf{D}_1$. The location of **P** in 3-space can be written as $\mathbf{C}_1 + r\mathbf{D}_1$, where *r* is the undetermined range parameter. Rather than solving for the direction vector $\mathbf{D}_2$ from the image of **P** in camera 2, (i_2, j_2), since the cameras are displaced laterally it is easier just to plug the indeterminate expression for **P** into the equation for the horizontal (column) coordinate of its image:

$$j_2 = \frac{\mathbf{D}_2 \bullet \mathbf{H}_2}{\mathbf{D}_2 \bullet \mathbf{A}_2}, \text{ where } \mathbf{D}_2 = r\mathbf{D}_1 - \mathbf{C}_2 + \mathbf{C}_1$$

There is only one unknown, *r*, and the equation can be solved algebraically for the range from camera 1.

I would hasten to point out that it is not always easy to find the same target point in images from two different cameras. However, since the image of the ray $\mathbf{C}_1 + r\mathbf{D}_1$ (as *r* varies from 0 to infinity) in camera 2 is readily determined, the area of the image to be searched is gratefully constrained. Even so, the techniques of performing the match can become quite complex and will be the subject of future articles. Many methods rely on statistical matching of regions of the image around the target points, as in [2]. Another clever approach is to use just one camera and monitor the shift of the target in the image as successive pictures are taken while the camera is moved along a lateral track, as in [3]. Of course, a mobile robot may be able to accomplish the same result by watching the apparent motion of targets as it moves along.

Figure 6 shows a stereo pair of images taken by adjacent TV cameras having a baseline separation of approximately .5m. From examining the two views of the scene, it is apparent that the lateral shift of the position of a target point relative to the centerline between the two images increases the closer it is to the cameras, whereas a target further away is shifted less. It is this *parallax* shift that is the primary range clue in a side-by-side camera configuration such as this one.

The Advantages of a Structured Environment

Often it is possible to obtain range information by taking advantage of some additional constraint imposed on the robot's environment. If the objects or target points the robot looks at can somehow be confined to a known two-dimensional surface such as a plane, then the two equations provided by the image of the target are again sufficient for a completely determined solution.

Figure 7 (a) shows a camera positioned above a plane. If we know where the plane is in 3-space, then any point **P** in the plane can be described as $\mathbf{P} = \mathbf{P}_o + m\mathbf{X} + n\mathbf{Y}$, where $\mathbf{P}_o$ is the 3-space vector locating some point in the plane, **X** and **Y** are some coordinate axes defined in the plane, and *m* and *n* are the parameters used to define the coordinates in the plane of the point in question. Here again, if we see some target point that we know lies in the plane, the system reduces to two equations in two unknowns and is readily solved. (A straightforward exercise left the the enterprising reader.)

The greatest advantage is obtained when the camera is oriented so that the image plane is parallel to the object plane, as shown in Figure 7 (b). Here, since the camera's aiming vector **A** is orthogonal to the plane, the value of $\mathbf{D} \bullet \mathbf{A}$ computed from any point in the object plane equals the same constant: the height of the camera above the

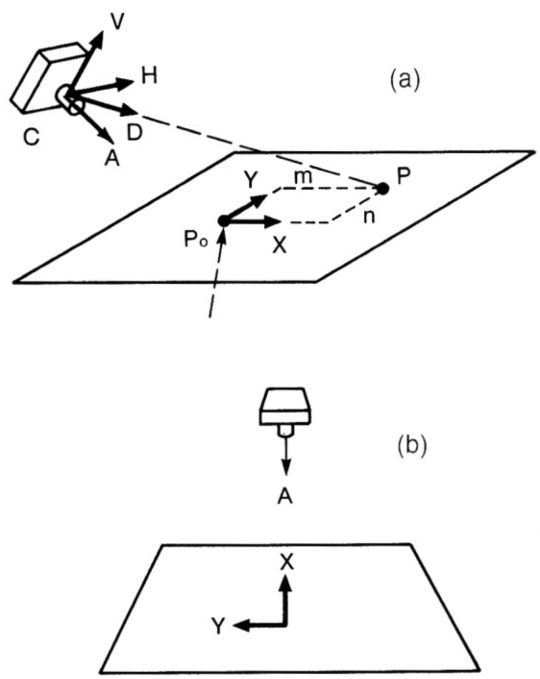


Figure 7. When target points are confined to a known plane, so that their location is completely specified by two coordinates (m, n) in the plane, there is a one-to-one correspondence between the location of points in the image and their position in the plane. In (a), the ray **D**, *computed from the image coordinates, intersects the plane at a unique, easily determined, m and n. When the camera is aimed* directly *at the plane* (b) *rather than obliquely, the mapping from image coordinates to plane coordinates is linear, and may be further simplified by aligning the axes in the plane with the image axes* **H** *and* **V**.

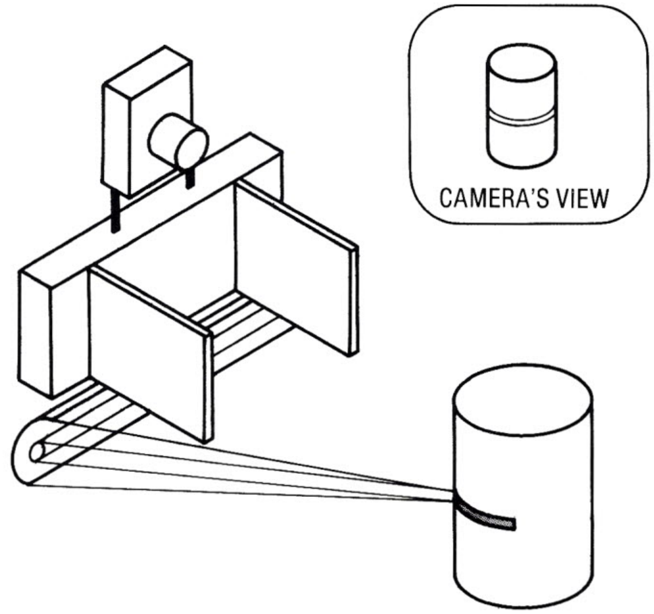


Figure 8. The effect of confining targets to a plane may also be obtained by projecting a plane of light, as done in this system developed at NBS. [5] *The timing of the flash is synchronized with the camera scan, so that it sees a bright band across the target. The location of the light plane is known, allowing range measurement to points on the target it illuminates.*

plane. Since the denominator of the perspective transform equations is now constant, the mapping becomes linear, and there is no shape distortion due to perspective. This permits the use of simple object recognition strategies based on matching silhouettes. Industrial robot vision systems based on this configuration are commercially available. [4, 5]

Constraining the environment isn't the only way of providing the structure necessary for monocular range measurement. At the National Bureau of Standards, researchers devised a light source using a flash tube and a cylindrical parabolic reflector that is capable of projecting a thin plane of light. [6] (Figure 8). With both the projector and a solid-state camera mounted on the robot's gripper as shown, the equation of the projected plane of light relative to the camera center frame is known. The triggering of the flash tube is synchronized with the vertical drive, so that the band of light visible where the projected plane intersects a target object is easy to spot. Since a point in the projected plane of light can be described with two coordinates, finding the location of its image permits accurate range measurement. Robot vision using this technique will soon be commercially available as part of a robotic welding system. [5]

The Problem of Camera Calibration

Our discussion so far has not dealt with the question of how to find the vectors **C**, **A**, **H**, and **V** that define the location, orientation, focal distance, and digitizer parameters for our camera. Clearly, it would be quite difficult to measure these values mechanically with enough accuracy to yield a camera model useful for automatic range measurement. In this section, I will describe a method that can be used to compute the camera calibration vectors from data collected during a training session in which the robot records the image coordinates of target points whose 3-space location is known.

All of our 3-space measurements, both for the locations of points in space and the components of the camera orientation vectors, must be made in a suitable external coordinate frame that is fixed relative to the camera. If the camera is stationary, then a coordinate frame attached to some reference point on the floor can be used. If the camera is mounted on a pan/tilt platform so that it can be aimed, then the calibration must be performed with the platform set in one position. Later, the pan/tilt platform itself can be calibrated so that measurements made in a camera frame defined on the platform can be transformed into some other frame as necessary, using the azimuth and elevation angles of the platform as parameters.

For a mobile robot, the camera calibration should be performed in a coordinate frame attached to a point on the body of the robot that has a fixed relation to the camera mounting. Once performed, the calibration will permit the

position of targets to be determined relative to the robot, so that knowledge of the robot's location and heading (maintained by its navigation system) can be used to refer the relative locations of targets into some stationary frame. Alternatively, the robot may use the relative locations of landmarks it sees to determine its absolute location visually, as in [3].

Let's consider the case of a stationary camera viewing a grid pattern placed on the floor, as shown in Figure 9. Using the stationary coordinate frame as a reference, the location of easily visible reference marks on the grid can be accurately specified. The first operation of the camera calibration program is to identify successive reference marks on the grid, recording for each mark m its image coordinates (i_m, j_m) and its location in 3-space $\mathbf{P}_m$. Thus, each reference point will have five numbers associated with it. It is *essential* that data be collected with the horizontal grid placed at two or more different elevations, or once with it on the floor and again with it on the wall, otherwise there will be insufficient data for a solution to be obtained. Likewise, the total number of data points, n, must be greater than the number of unknowns. To find the four 3-space vectors we seek, a miniumem of twelve data points should be taken, and $n > 20$ is preferable.

The object of the calibration procedure is to perform a least-squares fit of the parameters of the camera model to the observed calibration data. Before we can write a program to do this, however, we must first examine the system analytically and formulate a solution algorithm.

We want to find the calibration vectors that produce the best fit in the relations

$$i_m = \frac{\mathbf{D}_m \bullet \mathbf{V}}{\mathbf{D}_m \bullet \mathbf{A}} \quad \text{and} \quad j_m = \frac{\mathbf{D}_m \bullet \mathbf{H}}{\mathbf{D}_m \bullet \mathbf{A}}$$

$$\text{where } \mathbf{D}_m = \mathbf{P}_m - \mathbf{C} \text{ and } m = 1, \ldots, n.$$

If our system were perfect, then each of the 2n equations would hold exactly. Sadly (but not unexpectedly) this is not the case. Round-off and digitization errors, camera nonlinearities, etc., combine to prevent it. Therefore, we have to formulate an expression for the "slop" in the fit. Starting with the equations for i_m, we can write this as

$$S = \sum_{m=1}^{n} (\mathbf{D}_m \bullet \mathbf{V} - i_m \mathbf{D}_m \bullet \mathbf{A})^2$$

This is the sum of the squared error terms we want to minimize by proper choice of calibration vectors. The solution to this system can be obtained by applying

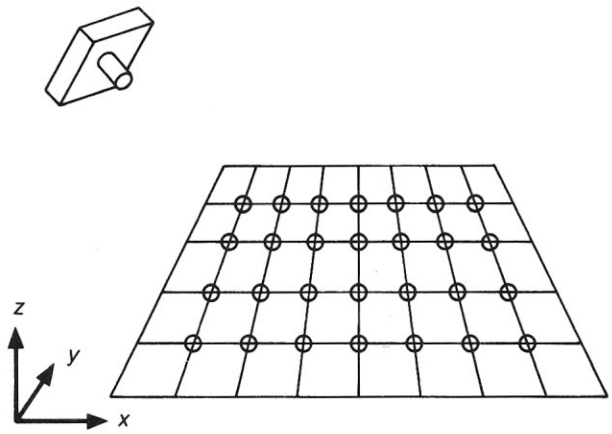


Figure 9. Camera calibration given a coordinate system defined on the floor (or wherever), the 3-space coordinates of reference marks on the grid may be measured and associated with their corresponding images as seen by the camera. Then, a least-squares fit of the camera calibration vectors may be obtained.

standard linear algebra techniques, provided it is rewritten in the correct form. Let's call the individual error terms inside the parenthese e_m. We can rewrite each e_m as

$$e_m = (\mathbf{P}_m \bullet \mathbf{V} - i_m \mathbf{P}_m \bullet \mathbf{A} - C_V - C_A), \text{ where}$$
$$C_V = \mathbf{C} \bullet \mathbf{V} \text{ and } C_A = \mathbf{X} \bullet \mathbf{A}.$$

Now we can more easily see the system as a set of n linear terms in 8 unknowns, 3 unknowns each for $\mathbf{V}$ and $\mathbf{A}$, and two more for C_V and C_A. The next step is to recall that our choice of $\mathbf{A}$ as a unit vector was arbitrary. To aid the solution, we can asume that one coordinate of $\mathbf{A}$, say $\mathbf{A}_1$, equals 1. This reduces the number of unknowns in each term to 7, in place of the eighth a constant term $i_m P_{m,1}$. We can always scale the solution for $\mathbf{A} \bullet \mathbf{A} = 1$ later if desired.

If we think of each term e_m as an element of an n-dimensional vector $\mathbf{E}$, then we can write an expression for $\mathbf{E}$ as an n by 7 matrix $\mathbf{Q}$ times a 7 by 1 column vector $\mathbf{W}$ containing the unknowns, with an n-vector $\mathbf{B}$ for the constants:

$$\mathbf{E} = \mathbf{QW} - \mathbf{B},$$

and the expression for the slop can be rewritten as

$$S = \mathbf{E} \bullet \mathbf{E} = \|\mathbf{QW} - \mathbf{B}\|^2$$

where the bars indicate the magnitude of the n-vector. Expressed this way, we are looking for the 7-vector $\mathbf{W}$ that minimizes S. As described in [2], the expression for the error n-vector $\mathbf{E}$ can be converted to a standard 7 by 7 inhomogeneous linear system whose solution is the calibration vector $\mathbf{W}$ that we seek. The conversion is accomplished by multiplying by the transpose of the

coefficient matrix and forming the expression

$$\mathbf{Q}^T\mathbf{E} = \mathbf{0}, \text{ or } \mathbf{Q}^T\mathbf{Q}\mathbf{W} = \mathbf{Q}^T\mathbf{B}.$$

Linear systems in this simple form can be solved by a variety of means: Cramer's Rule, Gaussian elimination, the Gauss-Seidel interative method, etc. The latter two approaches are more suitable for computerized solution; flow charts and FORTRAN programs for these are available in [7].

At this point, we know **A**, **V**, C_A and C_V. Knowing the values for **A** and C_A permits us to build an n by 4 expression for the slop in the fit for the n column coordinates j_m. The 4 unknowns are C_H and the three components of **H**. If you write the solution procedure to operate on any two dimensional array, then the same procedure may be used to solve both systems. (You may have to write your own array access functions if the language you used does not permit passing arrays of variable dimension to subroutines.)

After we have C_A, C_H, C_V, the 3-space location of the camera, **C**, is obtained by solving the 3 by 3 linear system

$$\mathbf{C} \bullet \mathbf{A} = C_A$$
$$\mathbf{C} \bullet \mathbf{H} = C_H$$
$$\mathbf{C} \bullet \mathbf{V} = C_V$$

and our model of the camera is completely determined.

It's a good idea to substitute the results into the original expressions for the slop of the fit to see how good your solution is. If the predicted location of the image of a known reference mark is more than a few pixels away from where it actually appeared in the image, then the solution is not accurate and should be repeated with a better set of data. The broader the distribution of your data points in 3-space, the more likely a good solution will be obtained by this method.

If your program should fail to obtain a solution (overflow, underflow, divergent iterations or whatever), then perhaps your implementation lacks precision or your original data collection was insufficient. In the former case a reformulation of your arithmetic expressions may prevent the loss of information due to round-off errors, while in the latter you should collect a new set of calibration data with a broader spatial distribution. Another potential problem is if the arbitrary choice of $A_1 = 1$ is inconsistent with your choice of the external coordinate system, i.e., that $A_1 = 0$ in that frame for your camera mounting. Without moving the camera, this can be remedied by redefining the correspondence between the 3-space coordinate indices and the actual x, y, z axes.

Concluding Comments

Although big "number-crunching" programs like the camera calibration procedure described here can place a burden on a microcomputer based system, the perspective transform requires only a few floating point operations to perform. Also, you should only need to run a calibration when the camera position or some digitizer characteristic is changed. Analog drift in the scan circuitry of a vidicon camera may necessitate more frequent calibration runs, however.

With the calibrated camera model, your robot vision system is equipped to make accurate measurements of spatial relationships in the surrounding world. Although a more complete presentation of binocular vision and other generalized range measurement methods must be delayed for future articles, the tools provided here are an essential foundation, and serve quite well in environments where targets are constrained to a two-dimensional space like a plane. R

References

[1] Alan Thompson, "Introduction to Robot Vision," *Robotics Age*, V. 1, n. 1, Summer 1979. (reprints available)

[2] Y. Yakimovsky and R. Cunningham, "Extracting 3-D Measurements from Stereo TV Cameras," *Computer Graphics and Image Processing*, No. 7, 1978, Academic Press.

[3] Hans Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," Report CS-80-813, Dept. of Computer Science, Stanford University.

[4] VS-100 Vision System, by Machine Intelligence Corp., Palo Alto, Calif.

[5] Autovision I and Robovision II, by Automatix, Inc., Burlington, Mass.

[6] R. Nagel, *et al*, "Experiments in Part Acquisition Using Robot Vision," National Bureau of Standards Report MS79-784.

[7] S. D. Conte, *Elementary Numerical Analysis*, McGraw-Hill Series in Information Processing and Computers, New York (1965).

TIG Welding with Robots

New techniques are leading to the automation of Tungsten Inert Gas (TIG) arc welding.

by
John Meacham
Vice President Engineering
System Technika

Introduction

In arc welding, heat from an electric arc fuses metals together. In gas tungsten arc welding (GTAW), the arc emanates from a nonconsumable tungsten electrode held above the workpiece. A blanket of inert gas (usually helium, argon, or a mixture of the two) protects both the tungsten electrode and the weld metal, since exposure to air would quickly oxidize them. People often refer to the GTAW process by its popular nickname, TIG (tungsten inert gas) welding.

Let's take a look at how a human performs TIG welding. Before welding, he cleans the workpiece of all contaminants, such as rust, dirt, oil, grease, and paint. The welder then strikes an electric arc—perhaps by quickly tapping the electrode on the work. Once the arc is lit, the welder moves his torch in a small circle until the heat creates a small pool of molten metal. When the welder gets adequate fusion at one point, he moves the torch slowly along the seam between the parts to be welded, melting their adjoining surfaces, and feeding the welding rod into the pool of molten metal, just ahead of the arc.

The welder must painstakingly control his welding speed, the speed he feeds the welding filler, and his welding current. He moves the welding rod and torch smoothly forward, making sure that the hot end of the welding rod and the hot solidified weld are unexposed to contaminating air. With foot controls, the welder adjusts the current to get proper fusion and penetration in the weld. He judges how much current to apply by the size of the molten metal puddle.

Manual TIG welding is a discipline that demands skill and experience. There are not enough TIG welders to meet industry's need for them, and, naturally, the welders are paid highly for their time. The power-generating, chemical, petroleum, and aerospace industries—all of which use high quality TIG welding—would all benefit from the automation of the TIG process.

TIG in Fixed Automation

Before we can understand how TIG is automated, we must understand the four process variables that determine the quality of the weld. These are: welding current, arc voltage, travel speed, and filler-feeding speed. The changes in these variables, taken over time, defines the *weld profile* for the TIG welding process.

To a large extent, welding current determines the quality of the weld. When we reduce the current, we reduce both the penetration and the width of the weld. For a given workpiece geometry, we usually want the current

to be held constant. When we need more than one current level during a weld, the current should "slope" up or down to the new level, rather than discontinuously jump from one level to the next.

If current is constant, we can measure the distance from the electrode to the workpiece (arc gap) as a voltage potential called *arc voltage*. In some welding systems, arc voltage is used for feedback to control the arc gap.

For a given current and arc voltage (arc gap), travel speed determines the amount of energy delivered per unit length of weld. In this way, travel speed affects the quality of the weld. Increasing the speed, while keeping current constant, reduces both the penetration and the width of the weld.

Finally, the welder needs to control filler-feeding speed—how quickly he adds welding wire to the molten weld. One method, commonly used in surfacing and in making large welds, is to feed filler metal continuously into the molten weld pool by oscillating the welding rod and the arc from side to side. The welder always keeps the welding rod near the arc, feeding into the molten pool.

In the simplest kind of automation, an operator sets the welding parameters to fixed values. The welding machine follows the preset values, with feedback on travel speeds and current. This kind of operation is known as fixed automation, and in TIG welding it is commonly used to join two cylindrical tubes together. A fixture clamps the two tubes together. The electrode moves over the seam between the tubes, welding in an "orbital" path around the tube, keeping the welding parameters constant. (It did not take much imagination to call this "orbital tube welding.")

TIG Welding with Robots

It is the tube's simple geometry that makes it so amenable to fixed automation. For a tube, we can keep the travel speed constant. We need only weld around the circumference. Since the thickness of the tube stays constant, we need just one current level, sloping up to the current level at the start of the weld, and sloping down at the end. To keep arc voltage constant, we need only keep the electrode at a fixed distance from the weld.

With geometries more complicated than a tube's, we need greater positioning precision. We also need a programmable weld profile—one that describes the complicated changes in current, arc-voltage, travel speed, and wire-feed rate at all times during the weld. To obtain positioning precision, we can use a robot arm, perhaps with an extra control for moving the electrode. To obtain a programmable weld profile, we need a programmable welding power supply.

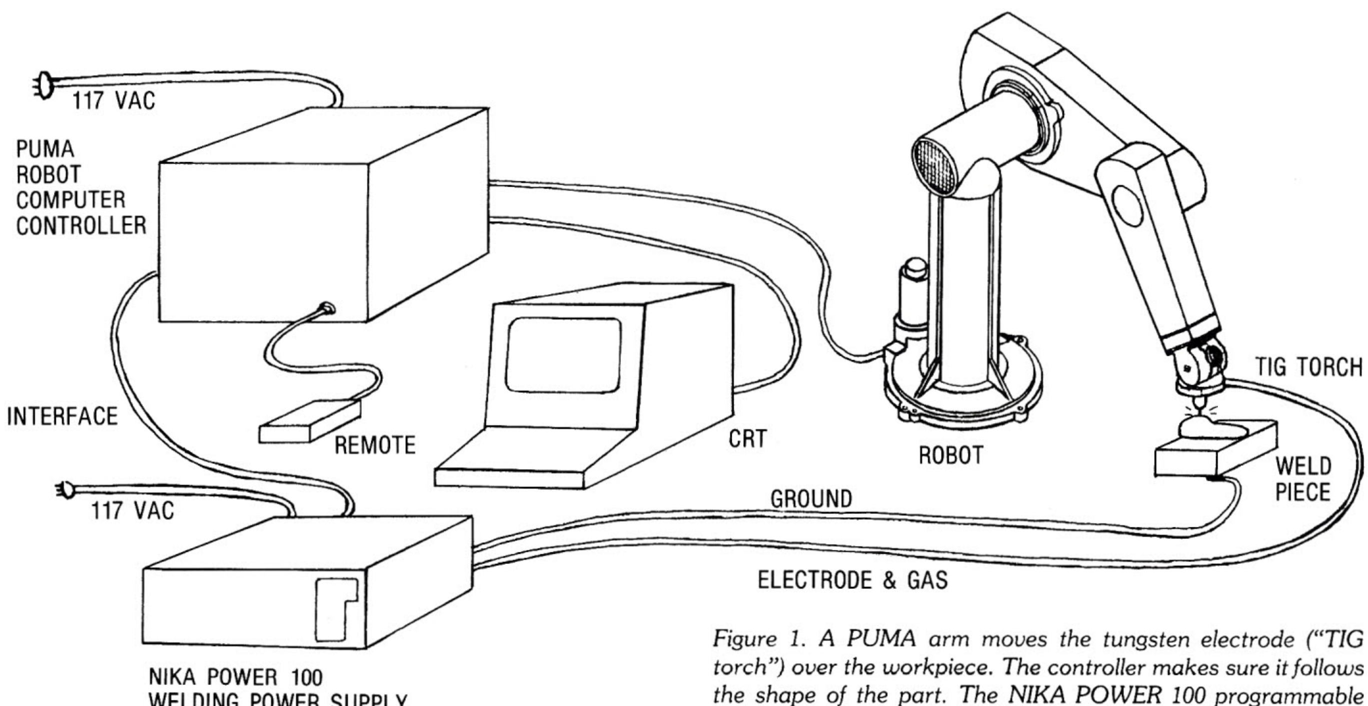
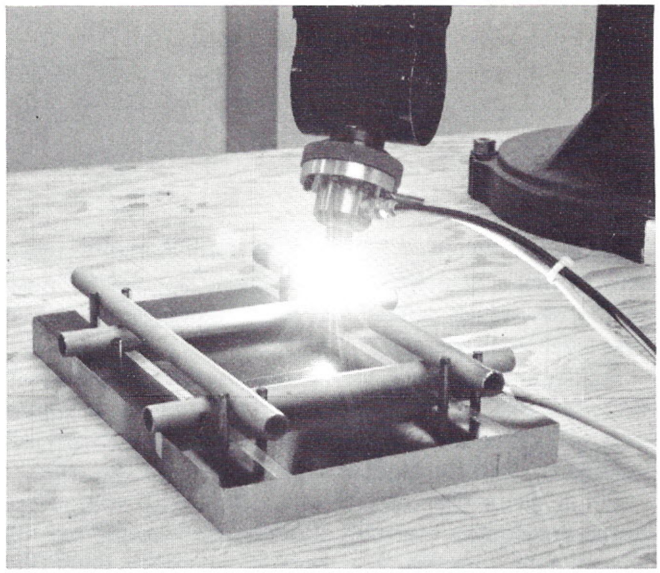


Figure 1. A PUMA arm moves the tungsten electrode ("TIG torch") over the workpiece. The controller makes sure it follows the shape of the part. The NIKA POWER 100 programmable power supply controls the weld parameters.



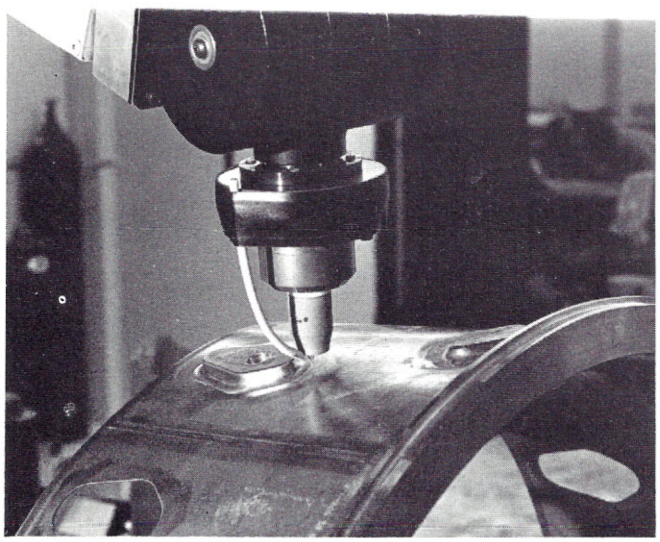
TIG welding with the PUMA.

At System Technika, we developed a system that interfaces a robot arm with a programmable welding power supply. Figure 1 diagrams the major components of System Technika's welding system. A robot arm (PUMA and modified Seiko arms have been used) runs the TIG torch over the workpiece, as a controller makes sure it follows the shape of the part, and as our NIKA POWER 100 power supply controls the weld parameters. Let's examine how this system performs a typical TIG weld.

During a set-up test run, the "welding engineer" programs the weld profile. In simplified terms, he sets the current (75 amperes would be a typical value), arc voltage (8–10.5 arc volts is typical with argon as the shielding gas), welding wire-feed rate (20 inches per minute is typical), and travel speed (5 inches per minute is typical). The welding engineer can program a different set of process variables for every time interval during the welding process. He can program more than a hundred points to hold different weld parameters. At this time, he also "teaches" the robot arm the shape of the workpiece.

Once programmed (taught), the power supply keeps welding parameters at their correct values for the appropriate portion of the welding sequence. The robot arm produces continuous path motion, traveling over the length of the seam.

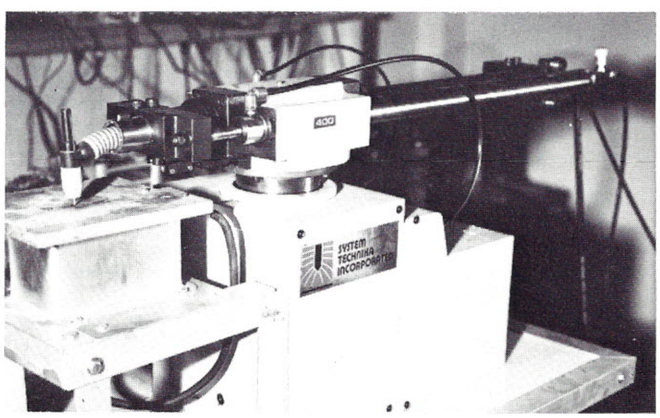
A human welder would adjust the current to get the suitable molten weld puddle size. Since the system cannot measure puddle size, it measures arc voltage, which (given constant current) is a function of the *z-axis* distance between the electrode and the workpiece. The further the electrode travels from the workpiece, the greater the arc voltage, the wider the arc gap, and the less heat penetrates the weld. The potential voltage drop across the arc is fed back to the NIKA power source, which then controls a z-axis motor on the end of the robot's arm. In this way, the system maintains the required arc voltage.



A PUMA arm positions the TIG welding torch over a workpiece.

The system also monitors itself for faults detected during the weld. If gas flow is interrupted at any time, the system immediately signals the operator, stops all motions, extinguishes the arc, and terminates the welding sequence. The system takes similar measures if it discovers that the workpiece is improperly connected to the power source's ground connector, if the liquid coolant for the torch head circulates too slowly, or if the torch accidentally touches or strays too far from the workpiece.

We have the option of programming the system for special welding techniques, such as oscillation and weaving. The system allows us to include weaving and oscillation



A modified SEIKO arm, set up for TIG welding. Notice that the z-axis distance between the electrode and the workpiece is controlled by a special z-axis motor.

speeds in the "travel speed" parameter. The weld engineer can program multipass welding by storing more than one set of welding parameters. While the system repositions the electrode for a second pass, it sets a "pilot current," as opposed to the normal welding current.

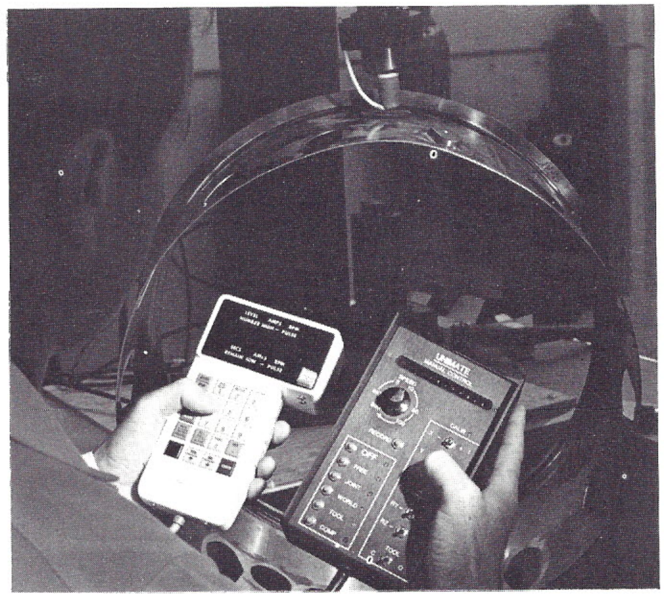
Conclusion

As we have seen, our current system has adaptive feedback in one direction—along the z-axis, perpendicular to the workpiece. In the near future, we hope to add feedback in the x-axis and y-axis directions as well, to keep the electrode tracking along the seam.

Seam tracking requires more sophisticated sensing. Eddy current sensors, sonic sensors, infrared sensors, and computer vision are all being developed. In addition, vision systems are being developed that can monitor the size and geometry of the weld puddle—information available, up to now, only to a human welder.

TIG welding was once possible only by using highly skilled, hard-to-find welders. Now, precise computer-controlled robots, working interactively with computer-controlled welding power supplies, have given industry the ability to automate the TIG welding process. Robots repeat their programs reliably, time and time again. With adaptive feedback correcting for variations in part fit-up, the robotic welding system welds as well as the best welder on his best day, day after day. Robotic welding amplifies the productivity of each *human* welder—who can now teach and supervise robots instead of doing all the *repetitive* welding by hand.

The System Technika NIKA POWER 100 programmable welding power supply.



Industrial Editor Jerry Saveriano displays the hand held teach pendants for the welding power supply (left) and the PUMA robot (right).

Golem's Oracle

by Jim Thomas

It squats and speculates
In future tense
Guzzling endless holes
Squeezed from subject's sense.
To search to sift to sort—
Swift as a wizard's wink.
To sever further audience
And seize eons to ponder —
To think.

Ah!
Let one, sparkling, scarlet ruby
not timely jot,
Causing the nourishing nectar,
to be naught.
Let the mimic's mind, airily dabble,
And observe . . . observe . . . observe . . .
The Boolean babble!

by
Martin Bradley Weinstein

ROBOT "DIGESTIVE TRACT"

An obedient, semi-intelligent charger and monitor for gelled electrolyte batteries.

Those of us with an interest in experimental mobile robots and similar mechanisms have by and large settled on batteries as the only practical means of providing power for the beasties. My own preference is for gelled electrolyte batteries, and the circuit described here is intended specifically for use with a Globe-Union type U-128 Gel/Cell®, a 12 Volt 28 Amp-hour battery; nevertheless, it should prove satisfactory for use with many other batteries, including both automotive lead-acid types (as well as those of other chemistry) and gelled electrolyte batteries of different capacities. In any case, the basic method is appropriate, even if component values must change.

I describe this circuit as "obedient" or semi-intelligent because it contains no decision-making circuitry and is designed to both supply information to and receive instructions from a controlling microprocessor. All input and output lines to the micro are 5-Volt logic compatible.

Operational Overview

The AC input to the charger is both bypassed and surge-protected, as well as protected by a self-resetting circuit breaker. The only element that is permanently connected to the AC input is a General Instrument Optoelectronics type MID-400 AC line monitor, a specially-configured optoisolator that maintains a low (open collector) output as long as suitable AC voltage is maintained with no dropout (low voltage) exceeding a couple of cycles. A pullup to a battery-derived regulated 5 VDC supply is provided as an ACOK output at pin 11 of the DIP connector.

Once the processor is satisfied that the charger has been plugged in to a "valid" socket, it can provide an active-high ACON signal to pin 9 of the DIP connector, which turns on a small 5 Volt DPDT relay (an ITT device,

available as part number 275-215 from Radio Shack), connecting the AC line to the primary of the power transformer.

The U-128 battery requires a maximum of 14.4 VDC at 6 Amps for proper recharging. The transformer you select will involve a number of trade-offs, including size, weight and cost. One easy choice is the commercially-available Triad F-257U, which provides 20 VCT at 6 Amps, which weighs in at 5.7 pounds and measures 3¾ (H) x 3⅛ (W) x 3⅛ (D). [1]

The geometry of my beastie required something shorter, so I decided to connect two 3 Amp transformers in parallel. This approach is acceptable only if the transformers are of identical manufacture, otherwise one of them may end up driving the other as a load, perhaps with unpleasant consequences. With this caveat in mind, two good choices are the Radio Shack 273-1514, 18 VCT at 4 Amps, which measures 4x2x2½; and the Digi-Key 308C, 22 VCT at 3 Amps, 2.6 x 3.1 x 2.7 [2], which is shown here. An additional word of warning: *be sure you wire the transformer secondaries in phase.* After tying one side of the two secondaries together, check to be sure there is zero AC voltage between the other two ends before connecting them, so that you don't short a series-aiding circuit!

The rectifier is a type PB-10 25 Amp 100 PRV Minibridge® by EDI [3], which is brought to a regulated 16.5 VDC by a National Semiconductor LM338 regulator—a nominal 5 Amp device that seems quite capable of delivering the 6 Amps peak current the charger requires. So far, pretty normal.

The next step allows processor intervention: a second connect relay controls the path between the output of the regulator and the balance of the circuitry. This relay is switched by a 2N2222 transistor, controlled by an active-high DCON signal at pin 10 of the DIP connector. A number of good choices for this relay are available inexpensively from Electronic Surplus, Inc. [4] The Sigma relay shown cost a whole dollar!

Now we get tricky again. Three series 3 Amp diodes are connected to four parallel 10 Ohm 10 Watt resistors (Radio Shack 271-132), and a similar arrangement parallel to the first. The result is a ballast resistor that both limits the maximum battery current to 6 Amps and provides a voltage proportional to the current through the resistors *added to* the 3 forward voltage drops of the diodes. Here's why.

The manufacturer's data on the U-128 indicates that charging is complete when current to the battery drops to somewhere between 180 and 460 mA, corresponding to a voltage across the ballast resistors of from 0.23 to 0.58 VDC. Added to the three diode drops, this is from 2.2 to 2.5 VDC. We can monitor this voltage handily with an optoisolator (the 4N33 in the schematic) to provide a recharge-in-progress RIP output at pin 14 of the DIP connector. This signal is low only when the charger is operating properly and the charging cycle has not yet completed.

DIP Connector Off-Board Interface Pinout

PIN 1:	Ground Reference
PIN 2:	"Hunger" monitor, *extreme emergency* level.
PIN 3:	"Hunger" monitor, *emergency* level.
PIN 4:	"Hunger" monitor, *extreme urgency* level.
PIN 5:	"Hunger" monitor, *urgent* level.
PIN 6:	"Hunger" monitor, *alert* level.
PIN 7-8:	Not used.
PIN 9:	CONNECT AC input.
PIN 10:	CONNECT DC input.
PIN 11:	ACOK output.
PIN 12:	Not used.
PIN 13:	DCON output.
PIN 14:	RIP (recharge in progress) output.
PIN 15:	Auxiliary +5 VDC output or reference.
PIN 16:	Auxiliary battery +V output.

The last bit of circuitry adapts a Texas Instruments type TL489 bargraph display driver as an independent state-of-discharge "hunger" monitor for the battery. With the 1N4735 Zener (6.2 Volts, Radio Shack 276-561) and resistor values shown, monitor points are provided for charge levels of 10.8, 9.7, 8.1, 7.0 and 6.0 VDC; a battery charge remaining of at least these voltages is indicated by a logic low at pins 6, 5, 4, 3 and 2 of the DIP connector, respectively.

Finally, ground and +5 VDC references are brought out at pins 1 and 16; pins 7, 8, 12, 13 and 15 are uncommitted.

You might have noticed the author's penchant for providing availability information and parts call-outs more often than you might expect in articles of this sort. This is to absolutely assure the reader's ability to reproduce the circuit shown with a minimum of frustration—and to circumvent Mr. Murphy's laws—*not* as a commercial forum for the products, manufacturers or distributors mentioned.

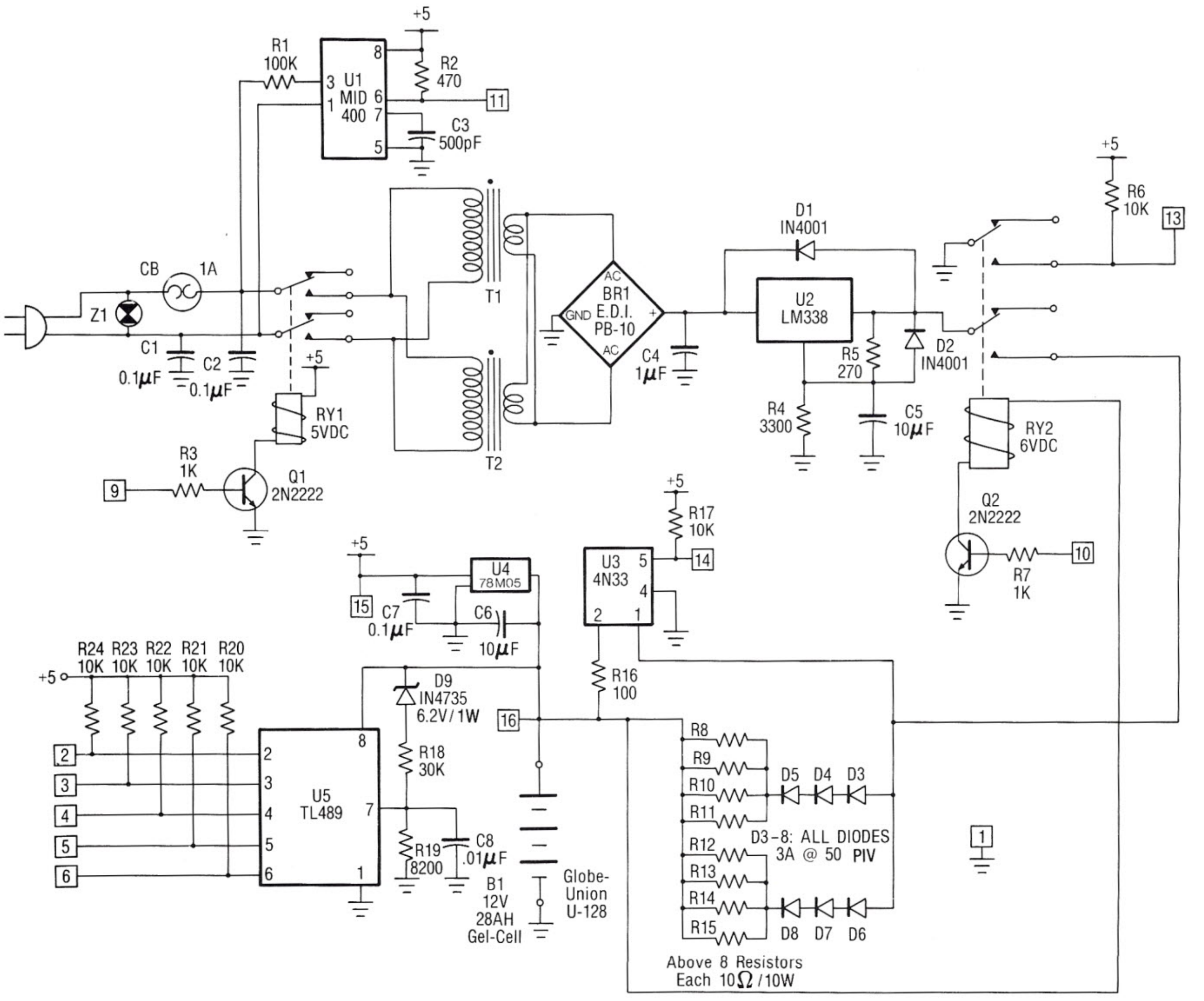
The AC Section

A GE surge suppresser and line de-glitcher is soldered directly across the connector terminals of the AC power-line (a standard tevision "cheater cord"), providing full-time service no matter how much of the remaining circuit is connected. A small 1 Amp self-resetting circuit breaker is wrapped with vinyl electrical tape and held in a vinyl cable clamp—a good practice with any glass-encapsulated device. A pair of 0.01 μF 200 Volt capacitors bypass RF and noise, not so much because any part of this circuit would prove sensitive to it but because general design practice supports using these cheap and lightweight ounces of protection.

At this point, the AC line is connected to both a double-pole DIP relay set of contacts and to a special optoisolator (U1), a General Instrument Optoelectronics (formerly Monsanto) MID-400 power-to-logic interface, used as a simple AC voltage monitor. With just three passive components added (R1, R2 and C3), this provides a 5 Volt logic compatible signal to pin 11 of the off-board DIP connector [see pinout listing].

The AC-connect relay is controlled by a signal at off-board DIP pin 9, which turns it on through Q1, a 2N2222,

The Robot Digestive Tract



with current limiting by R3. While 5 Volt logic is assumed throughout, even 12 Volt CMOS should drive it comfortably—providing a dedicated buffer or latch output is used. When energized, this relay connects AC power to the transformer primary (or primaries, as shown).

The Regulator

National Semiconductor has somewhat eased the task of designing a regulator capable of handling the battery's peak current demand: the LM338K, an adjustable 5 Amp (nominal, 6 Amp max) regulator does the job with a minimum of external components.

Globe-Union suggests the use of a voltage regulated, current limited recharger with their Gel/Cell batteries. The data sheet for the U-128 12 Volt 28 Amp-hour battery used here suggests a 14.4 VDC maximum charge at up to 6 Amps. Let's take a closer look at this requirement to see how that translates to particular parts values for the regulator's peripheral circuitry.

The schematic shows ballast resistors R8 through R15 and diode strings D3/4/5 and D6/7/8 in series between the regulator and the battery, and we need to sum the *end-of-charge* voltage drops across these elements with the 14.4 VDC the battery requires to determine the output

Parts List:

B1	Battery, 12 VDC 28 Amp-Hour Globe-union type U-128 Gel/Cell®
C1	.01 μF 200 Volt
C2	.01 μF 200 Volt
C3	500 pF
C4	1 μF tantalum
C5	10 μF tantalum
C6	10 μF tantalum
C7	0.1 μF
C8	.01 μF
CB1	1 Amp 125 VAC self-resetting circuit breaker (available from Electronic Surplus, ref. [4])
D1	1N4001
D2	1N4001
D3-8	3 Amp/50 PIV diode, 1N5400 or similar (available from Electronic Surplus, ref. [4], or as Radio Shack part 276-1141)
D9	6.2 Volt 1 Watt Zener, type 1N4735 (available from Radio Shack, part 276-561)
BR1	Bridge rectifier, 25 Amp 50 PIV unit used is EDI part PB-10 (see ref. [3]—actually, this is rated 100 PIV), or use Radio Shack 276-1185
R1-R7	¼ Watt, values as shown
R8-R15	10 Ohm 10 Watt wirewound (Radio Shack 271-132)
R16-R24	¼ Watt, values as shown
Q1, Q2	2N2222
RY1	5V DPDT relay (Radio Shack 275-215)
RY2	6-9 V DPDT relay (check with Electronic Surplus, ref. [4])
ST	Twin screw terminal strip [to connect battery] Radio Shack 274-663
SK1	16-pin low profile DIP socket (for RY1)
SK2	16-pin long-lead wire-wrap-type DIP socket (Radio Shack 276-1994), used to bring off-board signals to chassis level from circuit board
T1, T2	Power transformer (see text), 22 Volts at 3 Amps
U1	General Instrument Optoelectronic type MID-400 AC Line Monitor, power to logic optoisolated interface IC (available through Hamilton-Avnet)
U2	National Semiconductor type LM338K 5 Amp [nominal, 6 Amp rated maximum] adjustable regulator (also available through Hamilton-Avnet)
U3	4N33 Darlington output optoisolator (Radio Shack 276-133, which is closer to a 4N31, is fine, or check with Electronic Surplus)
U4	78M05 5 Volt 3-terminal regulator (or use a 7805 or LM340T-5.0 or Radio Shack 276-1770 or order through either Digi-Key or Electronic Surplus)
U5	Texas Instruments type TL489 5-step analog bar graph driver
Z1	General Electric V150LA20A transient absorber (available from Electronic Surplus)

References for Parts:

[1] Triad F-257U, Triad-Utrad Distributor Services, Div. of Litton Industries, 305 N. Briant St., Huntington, IN 46750.

[2] Digi-Key Corporation, PO Box 677, Thief River Falls, Minn. 56701

[3] Electronic Devices Incorporated, 21 Gray Oaks Ave., Yonkers, NY 10710.

[4] Electronics Surplus Incorporated, 1224 Prospect Ave., Cleveland, OH 44115.

Other Items:

- Terminal lug strips
- Printed circuit breadboard (Radio Shack Archer 276-170 in prototype)
- 1½ x 4½ x 8 inch chassis box (Bud AC-1407 or equivalent)
- Appropriate heat sinks (Digi-Key part 690-3B or HS110-3 with Aavid 5791, or equivalent)
- Chassis-mounted AC "cheater" connector
- AC "cheater" line cord

requirement of the regulator. The manufacturer specifies an end-of-charge current for the U-128 of 180-460 mA (substitute data for your battery). This current through the ballast resistors (composite parallel resistance of eight 10 Ohm 10 Watt resistors is 1.25 Ohms at 80 Watts) yields a drop of 0.225-0.575 Volts, which is added to the forward drops of the series diodes.

The bottom line is a requirement for something close to 17 VDC out of the regulator, accomplished according to the LM338 data sheet by 3300 Ohms at R4 and 270 Ohms at R5. The only capacitors needed are the 1μF at C4 and 10μF at C5, assuming you use tantalums. Multiply these values by 5 or 10 if you have to use standard electrolytics, and get ahold of the LM338 spec sheet for more information.

Diodes D1 and D2 provide reverse surge protection—an unlikely eventuality with this circuit, but again, a cheap ounce of protection.

The output of the regulator is connected *only* to the DC-connect relay, and is connected to the balance of the charger circuitry only when this relay is energized. The relay driver is a duplicate of that at the AC-connect relay, and is driven by an appropriate signal at pin 10 of the off-board DIP connector.

All About Ballast

Let's look at a couple of figures: what happens when the 6 Amp maximum current we talked about is flowing through the 1.25 Ohm ballast resistor? Obviously, there's a 7.5 volt drop. Now, turn this around. If there's 14½ Volts (more or less) on the supply side of the ballast, only a battery depleted to 7 Volts charge remaining would cause a 7½ Volt drop, and only then would the full 6 Amps be drawn through the ballast.

Practically speaking, we will want to have recharged the battery long before this. A 5 Volt regulator, for example, usually wants about 8 Volts at its input in order to function properly. The manufacturer doesn't talk about any life being left (although there's always *some*, practically speaking, with a gelled electrolyte battery, even after severe discharge) below 7.2 VDC.

If the battery is ever allowed to drop to 6 Volts remaining charge, this ballast resistance would mean a 7 Amp draw, which is beyond the rated capabilities of the transformers (depending on your selection) and the regulator, and some damage to the battery may have occurred. You take your chances recharging at this point.

So, recognizing a practical lowest discharge limit of 7 Volts at the extreme, the 1.25 Ohm ballast resistor provides practical current limiting within the manufacturer's suggested limits for the battery. With eight 10 Watt resistors in parallel (for 80 Watts capacity), this ballast would be comfortable with 8 Amps going through it, so it's very comfortable with the 6 or less it will encounter here. In any case, the resistors were mounted close to the chassis on terminal lugs, and placed in contact with the chassis with a layer of heat sink compound, helping radiate any heat that's produced.

The Ohm's Law performance of the ballast keeps current proportional to the voltage difference between the supply and the battery, with constant tapering as the battery charges up.

The Monitors

There are three functions being monitored by charger circuitry here and reported on appropriate pins of the off-board DIP connector.

The first monitors the status of the DC-connect relay through the contacts of a second pole. Pin 13 provides a 5 Volt logic compatible low when this relay is energized, a high otherwise.

Optoisolator U3 (a 4N33 here, but a 4N31 or something else could work as well—the Darlington ouput isn't an absolute requirement) monitors the voltage drop across the ballast resistors and associated diodes (in other terms, between the regulator through the relay and the battery). In fact, the only reason for the series diode strings is to increase this voltage drop to a point where it's more easily monitored by the optoisolator. The forward drop across the LED in U3 is 1.5 volts. With R16 (100 Ohms) to limit current through the diode, U3 is on as long as there's a charging current through the ballast resistors and diodes *greater than* the end-of-charge current, at which point the LED is quenched off.

So a logic low at pin 14 means specifically that a recharge cycle is in progress; a logic high means that either the charger isn't powered (which can be double-checked at pins 11 and 13) or that the end of charge has been reached.

U5 provides a battery monitor function that is incorporated both out of convenience and space availability—it could be implemented independently of the charger. It provides five logic outputs that correspond to specific levels of remaining battery charge.

An inexpensive bar graph driver is used, the Texas Instruments TL489, which is fundamentally five comparators in an 8-pin DIP. An expanded scale is provided (normally, the driver switches at 200 mV intervals from 0-1

Volt) by knocking the top 6.2 Volts off the battery voltage with Zener D9, then dividing what's left with R18 and R19.

Now let's see how the battery monitor corresponds to "real life."

The Battery Monitor

At what point does a battery need recharging? This depends a great deal on the type of application. If the powered device is a power-failure safety light, the number of choices is reduced. If it's involved in a safety task (say paramedical support equipment, or a rescue robot), its ability to complete the task on the charge remaining is an important consideration.

You'll have to answer this question for yourself, or provide software to determine it. The battery monitor here is designed to aid that decision.

Five specific "state-of-discharge" trip points have been established, above which the output on the corresponding off-board DIP connector pin is at a 5 Volt logic compatible low, below which it's high.

The highest of these, an "alert" level, corresponds to a 10.8 Volt trip level. This is a very comfortable discharge level, and the device may request recharging at its convenience once this level has been reached. The *need* for recharging is less than critical, and may be approached leisurely *if* the device is not engaging in a high-current-demand task. If a high-current-demand task is anticipated and this level has been reached, it would be preferable (but not mandatory) to recharge before engaging in the task. This signal appears on pin 6.

The next, at pin 5, trips at 9.7 Volts and corresponds to an "urgent" level. Unless the device is engaged at an urgent task, recharging should be considered mandatory at this point. A battery (similar to the U-128) delivering 15 Amps for an hour or 25 Amps for thirty minutes discharges to this level rapidly toward the latter moments.

At pin 4, a signal corresponding to an 8.1 Volt charge provides an "extreme urgency" warning. At this point, 5 Volt regulators are within a few tenths of a Volt of losing their grip, motor speed and power available are reduced, and almost all reserve capacity from the battery is gone.

Pin 3, a 7.0 Volt alert, signals an "emergency" condition. At this point, the recharger going full blast can still "save" the battery, though many on-board systems may have long since begun behaving undependably or erratically.

Pin 2 triggers at 6.0 Volts, signalling an "extreme emergency" condition. At this point, its best to shut all systems down, inspect them scrupulously and baby the battery back to health. At least temporarily, virtually everything has "had it."

It's helpful, as some of the popular books on hobby robotics suggest, to think of these levels of discharge as degrees of "hunger." The analogy is complete if we consider the battery to be the robot's "stomach" and this circuit its "digestive tract."

Plus Some Extras

For convenience, 3-terminal regulator U4 provides a regulated 5 Volts for on-board functions and off-board 5 Volt logic compatibility. This 5 Volts is also brought to pin 15 of the DIP connector for facilitating testing or powering (modest) connected controls. Ground is available at pin 1, and the full battery voltage at pin 16.

Connections to the battery are made through a two-screw terminal on the top of the chassis. Remember, this particular circuit needs to carry up to 6 Amps (normally, possibly a bit more), so be generous with the wire gauge you select for connecting the battery.

Conclusion

It doesn't take much to add a little sensitivity to the brute force approach of most battery chargers. Indeed, 6 Amp automotive-style chargers have been on sale for under twenty dollars recently, and these would easily accomplish the basic recharging job. But a dumb charger isn't the most intelligent choice, especially when it's going to be used with intelligent systems.

I would suggest that whatever controlling intelligence governs the device's behavior under critical conditions be CMOS, some other low-power technology, or independently or failsafe powered. The Motorola MC146805E2L is a CMOS version of the 6805, compatible with a large number of 6800-family instructions, and an excellent choice.

Checkout of the circuit is very straightforward. I used a Hickok MX-333 DMM with logic probe and audio monitor functions in conjunction with a Heathkit IP-18 variable supply and got the whole thing done while watching an episode of *Buck Rogers* (how about those arms on the Crichton robot!). The values indicated are the result of actual measurements on my prototype, and may vary with other component choices.

If you have refinements or modifications to suggest, please send them directly to **Robotics Age**. And more power to you!

Eye to Industry

by J. W. Saveriano

Newsnotes...In the last issue I gave **Metalworking News** as a good source of news about robotics and manufacturing. Another source that I find useful is **Industrial Robots International**, published monthly by Technical Insights Inc., with Kenneth A. Kovaly as publisher and Dora K. Merris as editor.

This publication is intended for those who need to know the latest developments in robotics. Since a single year's subscription costs $144.00 in the U.S., **Industrial Robots** is obviously for serious industrial roboticists only. Readers who want more information should send requests to P. O. Box 1304, Fort Lee, New Jersey 07024.

Here are a few of the stories that **Industrial Robots** covered in the February 1981 issue: **Advanced Robotics Corporation** (ARC) of Hebron, Ohio has added a new large welding robot to their line. The new robot has a reach extending two meters up and two meters out. It can be tracked to go as far as needed in the horizontal direction. In one application, it travels eighty-five feet. The robot also works on a swivel base, allowing it to reach parts on all sides. ARC programs both the robot's part-positioner, which holds the workpiece, and the arm that holds the welding torch. Because of this, ARC robots can reach positions that are difficult to weld.

Fujitsu Fanuc of Japan and **Seimens** of West Germany will jointly develop a line of assembly robots and the sensors that guide them. Intended for introduction in 1985, the robots will be able to work on a broad range of machines and equipment.

Trallfa robots, currently built in Norway, and widely used in spraypainting, will also be manufactured in the U.S., by **DeVilbiss Corp.** The capabilities of Trallfa robots will soon expand beyond spraypainting. DeVilbiss plans to add arc welding and material handling to its reportoire.

From other sources: **Heath** is building prototypes of an inexpensive robot that it plans to introduce in 1982 (see *Organizations*). I have also heard that the **Robot Mart** has opened the world's first retail "robot store" in New York City.

Cincinnati Milacron is preparing to announce new robots, all electric, both larger and smaller than their popular T-3 hydraulic arm. As one of the most successful suppliers of machine tools, Cincinnati has made a firm commitment to become number one in industrial robots. They are rapidly expanding manufacturing facilities and aggressively funding advanced robotic research and development. Some of their research is being conducted at UCLA—a long way from home.

Meanwhile, back at the Danbury, Connecticut home of **Unimation**, Ellen Mohr, a marketing and advertising executive, says that **Unimation** has no intention of letting Cincinnati usurp their position as industry leader. By this summer, **Unimation** has plans to expand their production from fifty robots per month to between eighty and one hundred robots per month. To do this they are expanding their 90,000 square feet of manufacturing facilities by adding 54,000 square feet in various Danbury locations. They are also increasing their workforce from five hundred to seven hundred employees to support the explosive growth in the robot market.

While the giants of the industry are digging in for combat, small and aggressive new firms are developing robots. New robot-makers are springing up everywhere—especially on the East and West Coasts, near major universities where they can gobble up new robotic talent before the ink can dry on new graduates' diplomas.

The push by new small companies to enter the expanding robot market brings an interesting question to mind: Can small and medium-sized businesses survive in a future dominated by highly automated, well-financed multinationals?

Can Small Companies Survive the Robot Revolution?

IBM, General Motors, General Electric, and other powerful corporations are now fully aware of the need to invest in productivity-boosting technology. Indeed, much of the research and development in robotics has been carried out or supported by such companies. We might expect, then, that these companies will be the first to benefit from robotization.

However, once the giant corporations have developed, installed, and proven a new technology, it "radiates"—first to the companies that work with and support the majors, and then to the small and medium-sized companies who must compete with the giants. Even the smallest of metalworking "job shops" now use numerically controlled (NC) and computer numerically controlled (CNC) machine tools. Thousands of small businesses have brought mini- and micro-computers into their offices—another example of an advanced technology that has radiated to small companies. Will this trend continue? And will it enable small companies to survive in the Robotics Age?

I believe that the answers are yes. Robotics will amplify

some of the small business's strengths, and even give it new advantages. For example: By the time small companies begin to use robots, they will be well-proven and affordable. It will cost the small companies less in time and money to change over to a new technology.

New companies just starting out can design their businesses to fully exploit the latest available tools. Older and larger companies, on the other hand, must overcome the inertia of the traditional manufacturing and business practices on which they are based. Not only is it difficult to change established practice, but it is also expensive to retrofit existing machinery with the new controls and other devices necessary for automation. Large companies will have to "rationalize" their manufacturing and office processes. But, as top management consultant Peter Drucker comments, small firms can start out right and "design (the business) on automation principles."

A small firm's management can make and carry out decisions quickly. Office automation, telecommunications, and management information systems will further increase its decision-making speed. A small company responds rapidly to an ever-changing market. It also incorporates product technology improvements into its manufacturing mix more rapidly. When sophisticated robots become affordable to small firms, allowing them to rapidly change parts or processes, they will amplify one of the small company's greatest strengths—its flexibility.

Economists commonly give several reasons to support the view of a future dominated by big business. First, they argue that tools such as computer aided design (CAD), computer aided manufacture (CAM), and robots require more capital than small companies can afford. Second, they assert that only a large company can finance the expensive research and development program needed to maintain the company's production position. The large company can simply outpurchase the small one.

However, if trends continue, the cost of robotic power, like the cost of computational power, may fall within the small company's means. These costs are reduced even further when the small company concentrates on manufacturing relatively "small goods" (a hand-held computer versus a truck, for example). Instead of spending money on long-term, high-risk R&D, and immediately attacking large markets, small companies can work on short-term, low-risk R&D, addressing the needs of (initially) smaller markets. For example, small companies can develop the software support for large scale integrated circuits, not the circuits themselves.

As society moves towards the postindustrial era, information and intelligence services will be the most rapidly growing market. Here too, the small company can benefit by concentrating on very specific and narrow market segments. If small businesses and individuals become expert at using computers and data bases, and tailor their services to consumer and industry needs, they will be more productive, pound for pound, than their larger competitors.

Advances in computers and software provide an opportunity to load more and more mechanistic record keeping and data processing jobs into intelligent machinery, thereby freeing individuals to do what they do best—creative thinking, and working with other people.

These developments may give rise to a whole new class of professional "micro-businesses," similar to the law, medical, and accounting consulting services available today. The growth-rates of the new micro-businesses will be comparable to the rise of the tradesman and craftsman classes in Europe in the fifteenth century.

The new micro-businesses may be as small as one or two individuals making use of the enormous power of computers, data bases, and advanced robotic manufacturing facilities from the individual's home. The new business class will be smart, not big. They will move information, not people or things. The trend in this direction can already be seen in some professions. Programmers, engineers, and consultants are working at home, using telecommunications to transfer data to their clients and customers. The "missing link" between CAD/CAM and robotics will rapidly augment these capabilities in profoundly new and unexplored ways.

The first industrial revolution changed the nature of man's work, moving him from the farm to the factory. Unfortunately, this move tied man to his machines. The next industrial revolution will free man from serving the machine. He will take his computing machines wherever he wishes, and use them whenever he wishes, providing goods as well as services.

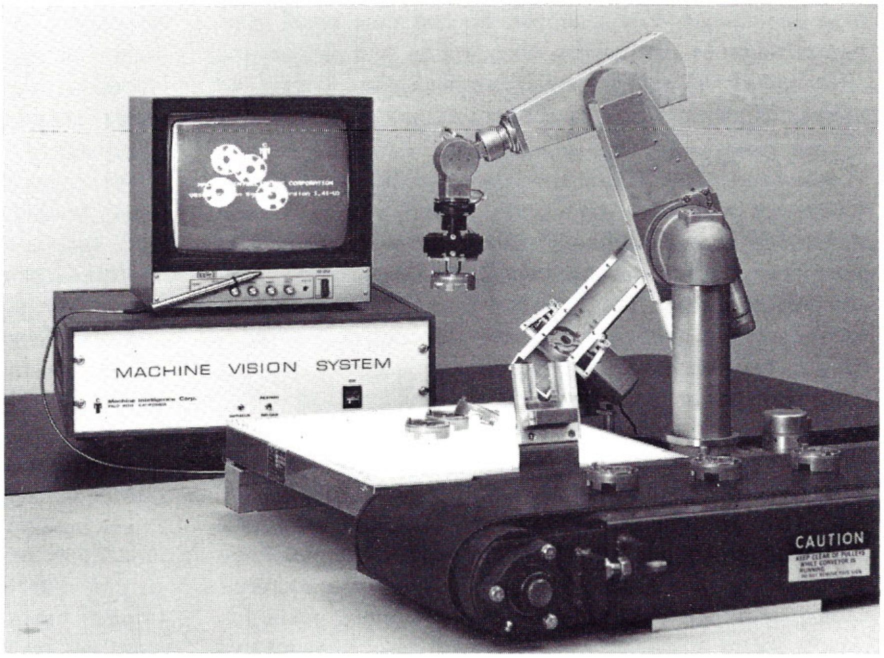
Can small and medium-sized businesses survive in the Robotics Age? Most assuredly! Not only will they survive, but they will *thrive* as machines shoulder the burden of the mechanics of business and production, freeing managers and workers alike for more creative effort. Small, talent-based companies may even help to redefine our concepts and methods of work.

The future belongs to the intelligent tool users. Those who learn how to use the powerful new tools of today will always have work tomorrow. *Be seeing you!*

If you have robotics-related information that you believe is noteworthy, or have suggestions for topics for this column, please write us at *Eye to Industry,* **Robotics Age**, PO Box 725, La Canada, California 91011, or call 213/352-7937.

NEW PRODUCTS

The UNIVISION™ System: PUMA™ meets the VS-100



The UNIVISION™ System is a complete robot/vision system comprised of a Unimation Inc. PUMA™ series manipulator and a Machine Intelligence Corp. VS-100 Vision System. The system is trained and operated through VAL™, Unimation's robot control language. It has the ability to determine the position and orientation of objects. In addition, the visual algorithms employed allow in-process inspection of workpieces in a material-handling or assembly system. The vision system is "trained by showing," that is, by taking pictures of sample parts.

There are now three machines in the PUMA™ series of Unimation Inc. manipulators. The Model 250 is a six axis machine with a 1 kg capacity, a reach of 0.4m, and tip speeds up to 3 m/s. The Model 500 and Model 600 are 5 and 6 axis machines, respectively, with a 3 kg capacity, a reach of 0.86 m, and tip speeds of 1 m/s. PUMA™ series manipulators are driven by DC servo motors with incremental optical encoders. Major axes have fail-safe brakes. Each joint has a two-stage, zero backlash gear train. The upper and lower link designs are of a monocoque construction which maximizes the strength-to-inertia ratio.

Each PUMA™ robot has its own dedicated computer controller. The heart of the system is the DEC LSI-11 computer. User programs are stored in RAM and the VAL™ software in EPROM. The servo system consists of a microprocessor and power amplifier for each joint. Each microprocessor receives a joint angle command from the LSI-11 and closes a software servo loop. This architecture allows dynamic updates of servo constants.

The teach pendant is microprocessor based and consists of joint drive switches, mode selection switches, a tool clamp switch, a teach speed potentiometer and an alpha-numeric display.

The overall success of the system depends on object recognition. The vision system characterizes blobs on the basis of such features as area, perimeter, minimum and maximum radii, and number of holes. To recognize objects, the vision system is "trained by showing" sample objects to the system. The user defines a prototype object by name and then shows it to the system in different locations, while the system accumulates statistics on the object's visual features. (*See the New Products section of Fall 1980 issue for a description of the Machine Intelligence's VS-100 vision system).*

The UNIVISION™ System combines the PUMA™ Series robots with the VS-100 Vision System. The vision system can process image data while the robot is moving, allowing typical cycle times of 3-5 seconds for acquisition and transfer of a part.

In the UNIVISION™ System, the vision system is slaved to the VAL™ robot control language. As a consequence, a number of vision related instructions have been added to the language. These instructions permit the user to calibrate the system, train vision prototypes, identify and locate workpieces, and store or load vision data from the Unimation floppy disk unit. During operation the robot can command that a picture be taken, and subsequently ask the vision system to locate a particular part. The vision system will process the blobs in the picture, and attempt to match them to prototypes. Among those blobs which match the prototype, the best match is chosen. If a match is found VAL™ defines a transformation with the same name as the requested part which

represents the location and orientation of the object relative to the camera reference frame, as determined by the vision system. If no match is found VAL™ can branch to another program step for remedial action.

The system can be set up and calibrated in a few minutes. The vision system can process image data while the robot is moving, allowing typical cycle times of 3-5 seconds for a part acquisition and transfer.

For more information, contact: Machine Intelligence Corp., 1120 San Antonio Rd., Palo Alto, CA 94303, phone—415/968-4008; or Unimation Inc, Shelter Rock Lane, Danbury, CT 06810, phone—203/744-1800.

CIRCLE 4

The GRIVET Series 5

The new Series 5, from The American Robot Corporation (TARC), is designed to manipulate objects weighing less than three pounds, while emulating human actions. The six-axis arm can reach a maximum of 32 inches and move its maximum load of 3 pounds at 3 feet per second with a repeatability of ±.004 inches. The GRIVET's controller is a single-board microcomputer with external data communication capability and diagnostic firmware. All instructions to the GRIVET arm are given through front-panel controls on the controller or through push-button commands on a hand-held teach pendant. For $10,000, the GRIVET is complete and ready to use. Contact: John Gallaher, The American Robot Corporation, P. O. Box 10767, Winston-Salem, NC 27108. 919/7888-8761.

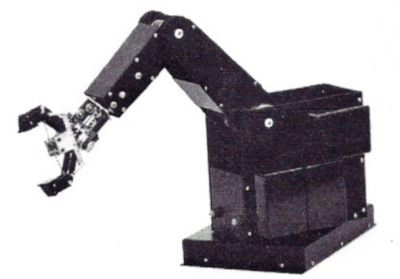
CIRCLE 5

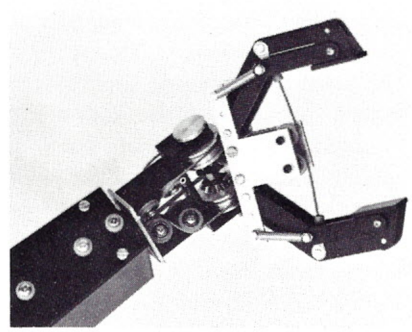
Sigma Introduces the "MAX" Manipulator

MAX is a compact robot manipulator arm with six degrees of freedom plus hand open/close. Positioning range is 20 inches (16 inches with the hand at 90°). The stepping motor drive assures accurate position handling of ±.010 inches per axis. Maximum load is one pound.

The hand has a 0 to 3 inch grasp. The six degrees of freedom are 180° clockwise/counterclockwise base rotation, 90° up/down lower arm movement, 135° up/down upper arm movement, 180° clockwise/counterclockwise wrist rotation, 90° up/down hand movement, and 180° clockwise/counterclockwise hand rotation.

The optional parallel ASCII interfaces control the stepping rates of the motor. Two interfaces are available: (1) a step-by-step interface that requires one character per step per channel, and (2) a vector interface that requires one character sequence which defines the total motion of each channel with execution time dependent on the component travelling the greatest distance. Both interfaces are compatible with printer interfaces to





most minicomputers and microcomputer systems. The vector interface reduces the work load imposed upon the computer.

Power requirements are +12VDC at 5A, and an optional power supply is available. Options also include a demonstration video tape and manual.

The robot manipulator arm sells for $2400, and options are offered at: step-by-step interface $195, vector interface $395, demonstration video tape $39.95, +12VDC, 5A power supply $119, and manual $19.95 (reimbursed at time of Max's purchase).

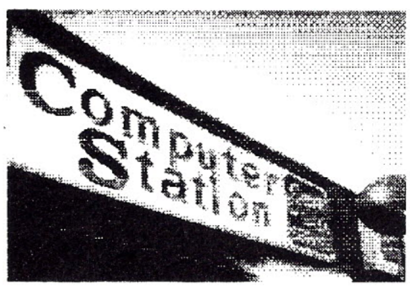
For more information contact Mary E. Atorthy, Sigma Sales, Inc., 6505 Serrano, Anaheim Hills, CA 714/974-0166.

CIRCLE 6

A High-speed, Low-cost Video Digitizer for Apple Computers

The Dithertizer II, offered by Computer Station, is a frame

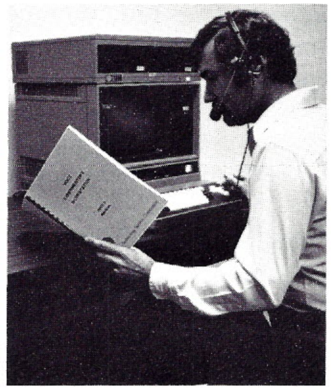
grabber, DMA-type video digitizer designed as a peripheral board for the Apple II computer. Using a video camera with external sync, Dithertizer II loads the Apple's high-resolution graphics area with a binary image. The image has a resolution of 280 × 192 pixels, and is captured in 1/60th of a second. Software, supplied with the board, alllows a user to choose the threshold of the binary image, build a "pseudo gray-scale" image from multiple binary images, and scan the image for the edges. Dithertizer II costs $300. A complete package, which includes a black-and-white Sanyo Video Camera, costs $650. Contact: Computer Station, 12 Crossroads Plaza, Granite City, IL 62040. 618/452-1860. **CIRCLE 7**



Voice Experimenter's Workstation

Consisting of two boards for insertion into an Intel Intellec microprocessor development system, plus associated software, the Voice Experimenter's Workstation allows the user to experiment with various isolated-word speech recognition, spoken phrase recognition, and computer voice response.

The first type of speech recognition handles words on short, run-together phrases which are separated by short pauses. The



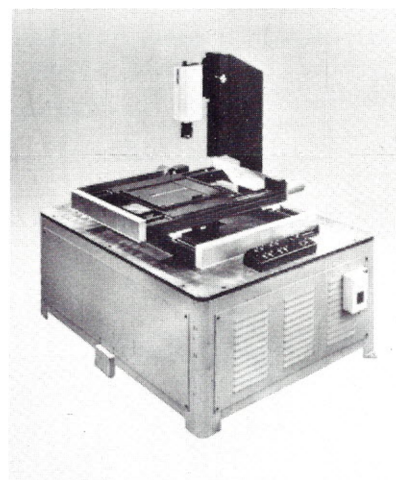
isolated-word recognizer is speaker-dependent; that is, the user must train the system to recognize his particular voice.

The Voice Control Unit employs the second type of speech recognition, discriminating between phrases or sentences chosen to be distinctly different. It is speaker-independent and capable of 99% or greater accuracy.

The voice response function stores and repeats phrases or sounds entered through the microphone. The phrases can be stored and accessed randomly. A menu-style software package is provided to allow the user to mix the two types of voice recognition and the voice response. TCS plans to make the Voice Control Unit available as a Multibus-compatible board for OEM's. The Voice Experimenter's Workstation will be available as a computer peripheral using a standard interface. The complete package sells for $7,500. Contact: Computer Sciences Division, Technology Service Corporation, 2950 31st St., Santa Monica, CA 90405. 213/450-9755. **CIRCLE 8**

EyeCom II Measures 35mm or X-Ray Film Images

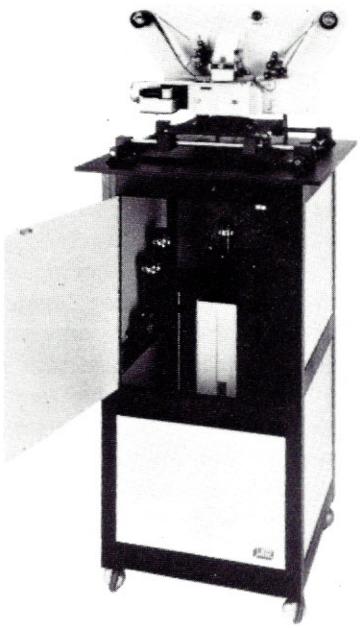
Spatial Data Systems offers an X,Y motorized positioning table for the EyeCom II to handle 14 × 17" x-ray film and dimensionally measure the image on the film using an automated edge finding routine. Calibration is incorporated in the image on the film to determine scale factor. The system has a light brightness servo system to control the amount of light incident on the lens regardless of the film density, over a range 0.3 to 4.0 density units. Accuracy of measurement depends on the scanner field of view. Resolution is approximately 1 part in 500 over the field of view. Table repeatabilty is ±.0005 inches.



X, Y Positioning Table

The Model 808VFS Pin Registered 35 mm Film Scanner is made to be used with the EyeCom II Picture Digitizer and Display System. The Film Scanner features four different magnifications with interchangeable lenses from 1:1 to 3.4:1. The 35 mm frame may be manually positioned relative to the optical axis of the scanner so that any portion of the frame may be viewed at any of the designated magnifications.

Applications for the system include the study and tracking of dynamic events, frame-by-frame, using digital image processing



35 mm Film Scanner

techniques.

For more information contact: M. S. Schlosser, Vice President, Spatial Data Systems, P. O. Box 978, 508 So. Fairview Ave., Goleta, CA 93017. 805/967-2383. **CIRCLE 9**

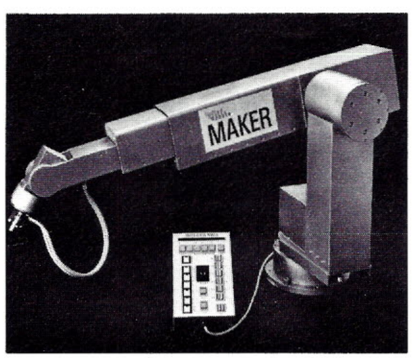
New Assembly Robot Features Telescoping Arm

United States Robots recently introduced a new assembly robot, the MAKER, designed to perform intricate assembly operations. The MAKER system consists of a five-axis arm, and a controller, and a hand-held teach pendant.

Four of the arm's five joints are rotary; but one is linear—a telescoping joint that extends from a retracted length of 16″ to 36″ when fully extended. This allows the MAKER to reach almost any point within a 72″ diameter sphere. The arm is constructed with high-strength, light-weight aircraft aluminum, and can smoothly lift 5 pounds at speeds up to 55 inches per second with a repeatability of .004″. The arm comes with a pneumatic gripper control. A light-weight, double-acting pneumatic hand is optional.

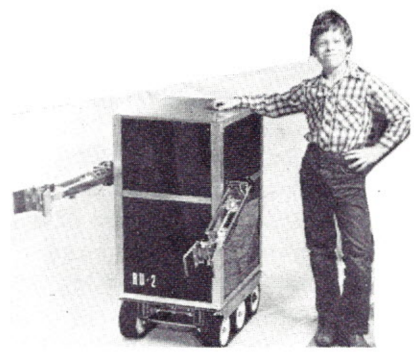
The arm is driven by a controller capable of storing 256 programs at one time. The microprocessor based design includes one microprocessor for each arm joint, plus one for mathematical calculations, and one which coordinates the other processors and performs system I/O. All processors communicate along a single bus; and the user can add optional modules. One option—remote diagnostics—allows system troubleshooting through a telephone link that connects the controller to a computer at United States Robots.

An operator can train the MAKER with its hand-held teach pendant. Through switches on the teach pendant, the operator can control any of the joints, command straight line motion between two points, edit programs, call subroutines, and request a system-status display.



Among the options for MAKER that United States Robots plans to introduce in 1981 are vision systems, force sensing, additional gripper designs, and bubble memory storage. Contact: United States Robots, 1000 Conshohocken Rd., Conshohocken, PA 19428. 215/825-8550. **CIRCLE 10**

The RU-II Robot from Hobby Robotics



Hobby Robotics Co. has announced the availability of their modules in a complete pre-packaged robot unit, the RU-II. The RU-II "Robot Unit-Series Two" is composed of the following modules: RBU-II Mobile Base Unit, SU-II Structural Unit (complete with SC-II Structural Cover, and CK-II Customing Kit), two complete AMU-II Arm Manipulator Units (each comprising Solenoid Actuated Hand Grippers, Motor Driven Forearm Unit, and Motor Driven Shoulder Unit), and three PCU-II Relay Power Units to control all motors and solenoids. The RU-II can carry an additional 50+ pound load of battery plus external load. Arm units are rated at over 15 pounds each. The RU-II is pre-assembled and knocked down for shipment. Minimal assembly is required. Delviery is from stock to 60 days, and the robot unit comes complete with the RU-II manual (and one year update) plus a free subscription to the Amateur Robotics Designer News Letter. The complete package costs $1495.00 (add 5% s&h). Contact: Hobby Robotics Co., PO Box 997, Lilburn, GA 30247. **CIRCLE 11**

Data Acquisition Module offers Digitally-Programmable Gain

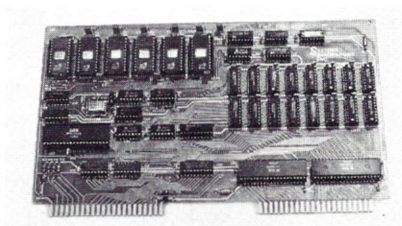
The DT5712, a new OEM-priced, 12-bit data acquisition module (DAM) featuring digitally programmable gain options and a jumper selectable input configuration of either 16 single-ended (SE) or 8 differential-input (DI) channels is now available from Data Translation, Inc.

The gain of a standard DT5712 can be fixed by a single, precision resistor over a gain range of 1 to 1000, with corresponding throughput rates of 40KHz and 3.8KHz, respectively. The input range spans a low level, 10mV for a gain of 1. With 12 bit resolution and ±½LSB linearity, the basic DT5712 delivers a relative system accuracy ranging from ±½LSB for a gain of 1 to ±0.1% for a gain of 1000. All versions can be user-connected to accept unipolar or bipolar analog inputs and to deliver the corresponding 12 bits of TTL-compatible, tri-state digital outputs in straight binary (unipolar) or offset binary and 2's complement (bipolar) data formats. Monotonicity is maintained throughout the 0° to 70°C operation temperature range. These units are fully microcomputer compatible and are utilized in several of DTI's single-board analog input systems.

For software programmable operation, the DT 5712 can be ordered with one of two on-card digitally-programmable gain options: the PGH option for gains of 1, 2, 4 and 8±0.05%, and the PGL option for gains of of 1, 10, 100 and 500±0.1%. The user can configure the DT5712 for either 16SE or 8DI input by jumper connections. The number of channels can be expanded to 64SE or 32DI channels by adding a compatible DT02EX DTI expander module.

Pricing for the basic DT5712 is $275.00 in 1-9 quantities. In the same quantities, the high level programmable gain (PGH) option is an additional $50.00 when ordering the DT5712; the low level programmable gain (PGL) option is an additional $100.00 (1-9) quantities). Contact: Nick Baran, Data Translation, Inc., 4 Strathmore Rd., Natick, MA 01760. 617/655-5300. **CIRCLE 12**

Industrial Single Board Computer



The CPU 65/08 SBC for general purpose industrial and commercial applications is available from Systems Innovations, Inc., Lowell, Mass.

Utilizing the 6502 Microprocessor, the board will accommodate up to 24K of ROM/EPROM and 8K of RAM.

Two on board VIA's provide 40 I/O lines including four 16 bit timer/counters, two serial lines and 14 levels of interrupt.

The I/O bus supports DMA and is fully buffered with pinouts equivalent to the KIM 4 standard, thereby allowing the CPU to drive expansion boards directly.

In small quantities, the unit is priced at $275.00 each and is available off the shelf.

For further information contact: Systems Innovations, Inc., N. R. Prevett, PO Box 2066, 505 Westford St., Lowell, MA 01851. 617/459-4449. **CIRCLE 13**

Fast 8 Bit A/D Converter achieves Ultra-Linear Performance

The ADC-881 is a high speed, successive approximation 8 bit A/D converter with a built-in sample-hold and circuitry for statistically improving the linearity. The converter uses the technique of averaging many analog steps to determine the ideal step size; systematic linearity errors are distributed in a pseudorandom fashion over the full range of the converter, thus appearing as noise rather than as linearity errors. By thus randomly distributing the errors, linearity is improved by a factor of 11.2 over its otherwise normal 1/4 LSB linearity; the linearity error is therefore reduced to .022 LSB, or .0087% for either integral or differential linearity error. This effect is particularly desirable in applications that use the digital output of the A/D to compile a histogram.

Other specifications of the ADC-881 include 5V analog input range, offset binary coding, an output overrange logic signal and 30 ppm/°C maximum temperature coefficient. Unaveraged RMS output noise is 0.2% of full scale. Power requirement is 15 VDC and +5 VDC at 6.6 watts. The compact case is black enamelled steel, measuring 5 × 3 × 0.375 inches (127 × 76 × 10 mm) and weighing 6.5 oz. (184 g.). The circuit is both magnetically and electrostatically shielded. The unit operates over 0°C to 70°C.

Pricing for the ADC-881 is $445. for 1 to 9 quantity and delivery is eight weeks, from Datel-Intersil, Sumner Eagerman, 11 Cabot Blvd., Mansfield, MA 02048. 617/339-9341. **CIRCLE 14**

MEDIA SENSORS

Business Week, Feb. 9, 1981, **From Japan, a Surprise Invasion of Robots.** In recent months, several Japanese firms have quietly established "beachheads in the U.S." in order to enter America's burgeoning industrial robot market. Japanese robot suppliers had previously hinted that their government wished to discourage exports in favor of improving Japan's own industrial productivity. Those hints have proven misleading. In reality, Japan was going through the preliminary phases of robot manufacturing—first, for internal use only, and second for limited export. Now, Japan is entering the third phase—aggressive export marketing.

Most American robot builders are small or medium-sized companies. Because robot sales have grown so rapidly (50% last year, to $90 million), these firms have been using most of their capital just to keep up with orders for existing products. They have little money left over for new product development. On the other hand, the Japanese robot firms are giant corporations. They already produce half of the world's computer-controlled robots, and they can afford to develop new products and to cultivate new applications.

Despite these advantages, Japanese companies must still overcome the problem of distributing and servicing their products in the U.S. As Joseph Engelberger, president of Unimation, observes: Japanese firms must "think about a license agreement, a joint venture, or setting up their own manufacture" in the U.S. Hitachi, for example, recently granted a license to Automatix, Inc. of Burlington, Massachusetts.

Hitachi's target is the fledgling arc-welding robot market. Most current welding applications are in the $30 million dollar spot-welding market, dominated by Unimation and Cincinnati Milacron. But by 1985, the arc-welding market may be worth as much as $150 million; and this market is coveted by the Japanese companies.

In addition to Hitachi, Shin Meiwa Industry Co. and Yaskawa Electric Mfg. Co. are competing for the U.S. arc-welding market. One Japanese industrial giant, Fujitsu Fanuc Ltd., plans to build robots from composite materials, rather than metal, to undercut domestic robot prices.

According to Ronald C. Reeve, president of Advanced Robotics Corp., American manufacturers see two strategies for fighting the competition: "Either you out-Japanese the Japanese with volume production or you find a small market niche they don't want." Advanced Robotics, among other firms (including established ones such as GE) are forming alliances with European companies in order to "out-Japanese the Japanese."

In the meantime, U.S. companies, which have had haphazard marketing strategies, are becoming more aggressive. As Laura Conigliaro, of the brokerage firm Bache-Halsey, observes, "the main thing that happened in the last six months is that U.S. robot companies have finally discovered marketing."

Infoworld, Jan. 19, 1981, **Micros Assist Disabled Vets.** Larry Leifer uses state-of-the-art electronics technology to benefit the physically impaired. Dr. Leifer is director of the new Rehabilitative Engineering Research and Development (RER&D) Center, part of Palo Alto California's Veterans Administration Medical Center. After studying the needs of people with physical disabilities, the RER&D Center develops and promotes aids to meet those needs.

The RER&D Center relies heavily on microcomputer technology, though it does not try to build "bionic" people. The bionic approach, Dr. Leifer feels, has led "into blind alleys time after time." "We have to focus on function," he asserts, "not anatomy."

Dr. Leifer is sensitive to "feedback" from potential users of his aids. He wants to avoid the kind of mistake made by designers of a hand-held laser-ranger, built to replace canes for the blind. The laser-designers failed to realize that when blind people tap their canes on a surface, they get more than simple rangefinding information.

The RER&D Center cares whether people feel comfortable about using their devices. Some people will accept a voice-controlled wheelchair; but Dr. Leifer worries

that others might think: "I'll be damned if I'm going down the street talking to my wheelchair. I'm an oddity as it is."

Quite a few devices at the RER&D Center originate in a "Smart Products Design" course, which Dr. Leifer teaches, as an assistant professor of mechanical engineering, at Stanford University. One of these projects, a robotic arm for people with manual dysfunctions, was created from a Unimation Electromechanical Arm, commonly used in industrial applications. A gripper on the arm consists of two finger pads, each using long-range and short-range photodiode sensors. For people unable to use keyboards or joysticks, voice-control is possible through a Centrigram Corp. voice-recognition unit. Two of Dr. Leifer's students, Charlie Wampler and Mitchell Weintraub, are now duplicating the arm for research at Stanford.

Another RER&D Center device is a smart wheelchair, developed by biomedical engineer David Jaffe. The wheelchair features two Polaroid ultrasonic transducers—from the sound-focusing cameras—which detect head movement. When the user tilts his head slightly forward, the wheelchair moves forward slowly, The further he tilts his head, the faster the wheelchair travels. A table, mounted across the chair's arms, holds other sensors. With these sensors, a user can command the chair to follow a corridor wall, or to put the chair under "cruise control."

Taking a standard electric wheelchair, Jaffe made it "intelligent" by adding a 64K RAM Zilog MCS 105 computer. The computer, attached to the back of the chair, is powered partly by its own battery, partly by the chair's. Before the chair can be used, its computer must be programmed from another Zilog. Loaded with a "training program," the second Zilog learns each user's range of possible head motions. Once the user has shown how far forward, backward, and sideways his head can move, the computer decides which head positions will represent which directions and speeds.

The Palo Alto RER&D Center—and similar groups in Chicago and Atlanta—have other projects in the works. These include a microcomputer based environmental control system for spinal-cord injury patients, and a more powerful speech synthesizer than the ones now available. Dr. Leifer encourages industry to suggest smart products that his students can design, preferring devices that are useful to the able-bodied as well as the disabled. The more broadly a product can be applied, the greater chance it has of being manufactured, and the less it will cost each user.

David Jaffe has remarked that rehabilitative engineers "don't want to make things too easy for the patients, because their self-esteem suffers." As Patty Winter, author of this article, concludes: "the real products" of the RER&D Center "are not electronic devices, but freedom and self-esteem."

UPI, **Robot Goes Haywire, Tears Self Apart in University Laboratory.** In a University of Florida Laboratory, an experimental robot went berserk. The automatic control system burned out a component and failed. Then, before graduate student Harvey Lipkin could press the cutoff button, the robot pounded its arm into its supporting stand until it tore off its shoulder. Fortunately, while the robot destroyed itself, Lipkin stood at a safe distance.

Besides Lipkin, no one witnessed the destruction—which took place at the University Center for Intelligent Machines and Robotics. For the past two years, the Center has been developing robots for industrial use.

American Metal Market, Jan. 26, 1981, **Robot Market Potential Lures New Manufacturers.** The robot industry projects that automobile manufacturers will use 50,000 robots by 1990. General Motors Corp. itself says it expects to use 14,000 robots by that year. Small firms specializing in robots are "springing up everywhere," hoping to grab a piece of this market. Recently, four of these firms described their products at a meeting of the Metro Detroit Robotics Chapter 235.

One of these companies is Bra-Con Industries. Specializing in transfer-machine type welders, Bra-Con sells mostly to the auto industry. Their recently acquired "Pacer" industrial robots are two and three-axis, computer-controlled, servo-hydraulic machines, available in five sizes, the largest of which has a total payload of 1,600 pounds. Selling through a subsidiary, Rob-Con Ltd., Pacer's prices range from $35,000 to $70,000 (less tooling and wrist for the two-axis machines). A third axis for swivel costs about $5000 more.

For over four years, four Pacer units have built millions of heater and air conditioning motors at GM's Delco Products division, Rochester, New York. Each robot averages less

than one hour a week downtime and costs an average of $500 per year to maintain.

Welding robots from another vendor, Advanced Robotics Corporation, use arc, TIG and MIG, though not resistance, welding methods. They are marketed to the aircraft, electronics, nuclear pressure vessel, and transportation industries. ARC has made several special welding systems built around their two five-axis robots, which use DC servo rack and pinion drives. Boeing has purchased two of these systems, using one in the assembly of the cruise missile. The firm also markets two standard models of welding robots—a one-half meter model, which costs $130,000, and a two-meter model, selling for $160,000. Thirty-three of these are now in use in the U.S.

Automatix, Inc. offers a five-axis robot with a fully-articulated arm. With an arc-welding package, the robot sells for $85,000. The plain robot has a 22 pound capacity and costs $81,000. The company also offers a programmable visual inspection systems. Able to inspect up to 360 randomly positioned parts per minute, the system can also be used as a vision sensory subsystem for industrial robots. It sells for $30,000.

United States Robots, headed by a former Unimation employee, is licensed by the Danbury, Conn. firm. Featuring a telescoping radial joint (*see New Products*), U.S. Robots' machine has no "elbow" to interfere with arm motion. The all-aluminum robot sells for $34,000. The gripper costs an additional $800, and a bubble memory backup sells for $2,000.

Toronto Star, Jan. 6, 1981, **He Has Seen the Future, and it is Robots.** Tough international competition, combined with slow economic growth, are forcing big governments and businesses to become more efficient by increasing automation. So asserts John Diebold, the man who coined the term *automation* years ago.

Diebold contends that some U.S. industries, such as automobile manufacturers, are in trouble because they spent less on research and development than other countries—especially West Germany and Japan. Two years ago, for example, Japan spent more on the development of assembly-line robots than all other countries combined. Japan has invested in "robotization," the new Diebold-coined word.

Business and government are listening to Diebold's assertions. The West German ministries of science and technology have long been clients of Diebold Group Inc. Diebold also advises some of the largest and most progressive Fortune 500 companies. The Reagan administration is seeking his advice on "non-military uses of computers in government."

A partisan of free-enterprise, Diebold insists that the government should not salvage Chrysler Corp. With proper "transition mechanisms," Chrysler and other "old industries should be allowed to die." New industries are being created by advanced technology. In the long term, labor should be diverted to these more supple and efficient industries. "Transition mechanisms" already developed in West Germany and Japan include better labor relations and education responsive to industry's needs. The U.S. should do the same, and stop "perpetuating the first half of the 20th Century's industries."

Toronto Star, Jan. 19, 1981, **Watchdog Robot Barks the Orders.** In his spare time, Donald Dixon created a look-alike robot that shames him out of midnight snacks, keeps his children in their rooms at night, and gives messages to his wife. Dixon, who named his robot "Ahmad," gave it a latex-rubber face resembling his own, with eyes that blink and lips that move when it speaks. It is microprocessor-controlled, with a drill-motor to turn the head, and three wheels on which to move around.

Dixon stations Ahmad by the door of his children's room. If they try to get out, Ahmad tells them to go back. If they persist, Ahmad awakens Dixon with an alarm.

Dixon views Ahmad as a prototype of the more advanced robots he plans to build when he gets financial backing. "In 10 years," he realizes, "robots will be as common in the home as microwave ovens."

Media Sensors *are brief summaries of robotics-related items that have appeared in the mass media. An attempt is made to paraphrase the content of the original item without altering the tone. The views expressed in these items are not necessarily those of Robotics Age. If you have an item you would like to contribute, send it along with a complete identification of its source, to:*

Media Sensors
Robotics Age
P. O. Box 725
La Canada, CA 91011

ORGANIZATIONS

SME Awards $136,200 in Grants

The Society of Manufacturing Engineers' Manufacturing Engineering Education Foundation announced the awarding of $136,200 in educational grants to 26 institutions.

"We received 73 grant proposals during the April funding period, compared with 67 during the first a year ago," said Foundation President Ralph Cross, Chairman of the Board of Cross & Trecker Corp., Bloomfield Hills, Michigan. "The Foundation recognizes the importance of this funding, and we see it as a major force in solving the present U.S. productivity lag."

Grants were made in five development areas: Capital Equipment, Student Development, Faculty Development, Curriculum Development, and Research. Nine schools received *Capital Equipment Grants* to cover the cost of start-up equipment and tooling, instrumentation and supplies.

Student Development Grants to fund scholarships, fellowships, co-op activities, graduate placement and innovative student development programs were awarded to ten institutions. *Faculty Development Grants* to support fellowships, travel subsidies and continuing education went to three schools.

Fourteen universities received *Curriculum Development Grants* for educational materials, teaching resources, faculty activities and travel. Four *Research Grants* were also rewarded.

The Foundation awarded $177,299 to 23 institutions last May, and announced that its original goal of raising $100,000 by November 1980 has been surpassed. Further plans call for awarding $1 million annually by 1982, SME said.

The Director of SME's Education Foundation is Dr. Warren W. Worthley, Indiana-Purdue University, 2101 Coliseum Blvd. East, Fort Wayne, IN 46805, 219/482-5327. The Manager of Fund Development is Richard Vogelei at SME Headquarters, One SME Dr., P. O. Box 930, Dearborn, MI 48128. 313/271-1500, ext. 511.

RIA Expands Engelberger Award and Invites 1981 Nominations

The Robot Institute of America (RIA) announces that its Joseph F. Engelberger Award has been expanded to three awards. The three award areas are Technology Development, Application, and Education. Presentations will be made at the 11th International Symposium on Industrial Robot (ISIR), October 7-9, in Tokyo, Japan.

"The field of robotics has expanded greatly and numerous individuals now contribute to it," said RIA President Stanley J. Polcyn, Vice President of Operations at Unimation, Inc., Danbury, Connecticut. "Our intent in diversifying this award is to give these specialists and their accomplishments due recognition."

Named for the founder and president of Unimation, Inc., and the first elected RIA President, the Engelberger Award has been presented annually since 1977. It is given for major contributions to advancing robots' service to man.

Originally the Award consisted of a commemorative medallion and a $1,000 honorarium. The three awards will each consist of a medallion and honorarium.

The RIA is presently accepting nominations among persons who have made significant efforts in the development of industrial robots.

The 1980 award recipient was Richard E. Hohn, Product Manager of Robotics for Cincinnati Milacron, Inc., Cincinnati, Ohio. Previous recipients include Ole Molaug, Research & Engineering Manager, Trallfa Co., Byrne, Norway (1979); Jerry Kirsch, President, Autoplace, Inc., Troy, Michigan (1978); and Prof. Yukio Hasegawa, System Science Institute, Waseda University, Tokyo, Japan (1977).

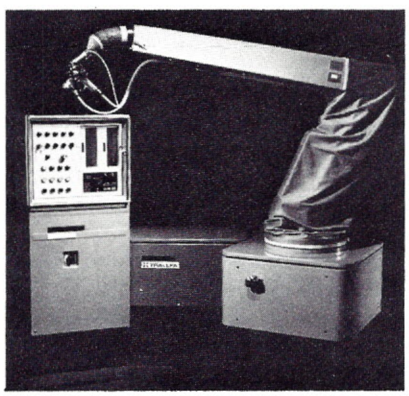
For additional information on the Joseph F. Engelberger Award and the nominations criteria, contact Lori D. Mei, Administrator of the Robot Institute of America, One SME Drive, P. O. Box 930, Dearborn, Michigan. Telephone 313/271-1500, ext. 407.

DeVilbiss to Manufacture Trallfa Robots

The DeVilbiss Company, a Toledo, Ohio-based manufacturer of coating application equipment and a

division of Champion Spark Plug Company, and the Trallfa Nils Underhaug A/S of Bryne, Norway, have signed a license and cooperation agreement which grants DeVilbiss exclusive North American manufacturing rights for Trallfa industrial robots.

This agreement sanctions DeVilbiss' use of all patents and know-how relating to the robots, which have been used extensively throughout Europe for both painting and welding. The DeVilbiss Company introduced the painting robot in the United States in 1975, and recently completed initial U.S. test marketing of the robot for welding applications.



In addition, a new long-term distribution agreement renews the 1972 accord which gave The DeVilbiss Company exclusive worldwide marketing rights with the exception of Scandinavia and Japan.

International Robotics Foundation

The IRF is establishing a worldwide network of clubs for amateur roboticists and students. IRF activities include a newsletter and regular meetings for home robot experimenters, a proposed contest for robot performance, and the $50,000 IRF prize for the first "intelligent" domestic servant robot.

The contest will test the abilities of robots in a variety of housework-related tasks. An IRF rules committee consisting of robotics experts is currently being formed to establish both the categories and tests for the contest and the requirements for the IRF prize.

For further information on IRF activities, the club in your area, or how to start one, contact Tom Carroll, president, International Robotics Foundation, 7025 El Paseo St., Long Beach, California 90815, or call 213/596-9769.

HEATHKIT to Offer Under $1,000 Robot

Though at least a year from retail distribution, HEATHKIT has told *Robotics Age* magazine that it plans to introduce a three-wheeled mobile robot. It will have manipulation, some on-board decision-making, and on-board sensing—for under $1,000. "It will most likely be based on the 6808 microprocessor," a company spokesman said, "Certainly on one in the 6800 family." There will be a sonar sensor on-board, similar to, but less expensive than the Polaroid model. For manipulation and mobility, the latest prototype has seven motors.

We were told that the prototype arm shows good repeatability, even after dozens of fairly complex operations. Because it must travel over floors of differing textures, the moving platform has a poorer repeatability. "Still," the spokesman said, "the capabilities offered for the price will surprise a lot of people."

The robot will operate in several modes, two of which are "automatic" and "teach/learn". In automatic mode, it will simply wander about, avoiding obstacles. In teach/learn mode, it will respond to commands from on-board computer programs.

HEATHKIT plans to market the robot primarily as an educational tool. With each robot will come printed course material, teaching modern industrial control techniques. Users will be able to wire-up their own experiments in an on-board "experimenter's area."

All the above information, HEATHKIT told us, is completely unofficial. The robot will most likely continue to evolve before the final design is chosen.

Robot Manufacturer Announces Equity Financing

Advanced Robotics Corporation, according to a company spokesperson, has acquired 2.4 million dollars in equity funds.

Company stock was sold to a syndicate of leading venture capitalists. Morgenthaler Associates of Cleveland and Golder-Thoma of Chicago are the co-lead investors. The investor group also includes Hambrecht & Quist, Northwest Growth and Interwest Partners.

The present focus of Advanced Robotics Corporation is on arc welding robotics. Robot vision, adaptive welding process control and sensor development dominate the company's R&D efforts.

Calendar

Management Update Seminar. The Society of Manufacturing

Engineers announces a "Manufacturing Management Update '81" seminar at the Hyatt House Hotel, Los Angeles, Calif., April 7-9, 1981.

Discussion topics will include CAD/CAM Perspectives, Graphics, Group Technology, Robots, Systems, Automated Assembly and Inspection, NC, Employee Relations in the '80s, Quality Circles, and Energy Conservation Techniques.

Seminar Leader will be R. William Campbell, President, R. William Campbell Associates, Wayland, Mass.

Registration fees are $450 for non-members, and $400 for SME and affiliate members. Certified Manufacturing Engineers and Technologists will earn 15 Professional Recertification Credits.

For additional information, or to register, contact James P. Lovell, Senior Program Administrator, Special Programs Department, Society of Manufacturing Engineers, One SME Dr., P. O. Box 930, Dearborn, MI 48128, 313/271-1500, ext. 391.

ELECTRONIQUE '81, the third industrial electronics exposition and conference organized to update Southeastern Michigan's diversified industries on new electronic product developments and applications, will be held April 21-23, 1981, at the Detroit Light Guard Armory, 4400 E. Eight Mile Rd.

ELECTRONIQUE '81 will explore components and microelectronics, EDP peripherals/data communications, instruments and control systems, and production packaging and test equipment.

Technical papers for the ELECTRONIQUE '81 conference are still being accepted. For details on presenting a paper, contact James Peirce, 23995 Freeway Park Drive, Farmington Hills, Mich. 48024, 313/477-7700.

For exhibiting information, contact Cindy Lowry, Exhibits Development Manager at Society of Manufacturing Engineers, One SME Dr., P. O. Box 930, Dearborn, Michigan 48128, 313/271-1500, ext. 310.

For other information, contact SME's Public Relation Department at 313/271-1500, ext. 323.

1981 International Tool and Manufacturing Engineering Conference and Exposition will take place at Cobo Hall in Detroit April 27-30. Twenty-four technical sessions, ten one-day symposia and four special clinics await the more than 25,000 manufacturing engineers and executives expected to attend from throughout the U.S. and several overseas countries. The event is sponsored by the Society of Manufacturing Engineers.

More than 450 exhibitors representing nearly 800 companies from 20 countries will demonstrate equipment, tooling and services in 85 categories, including machine tools, metalworking processes, tooling, accessories, and related production equipment, systems, materials and services.

Under its theme, "Reindustrializing for Higher Productivity," the SME Conference will focus upon "manufacturing's most significant advances and opportunities in today's highly competitive industrial world."

The one-day symposia wil focus on mechanical finishing, industrial painting processes, CAD/CAM, transfer presses, engineering fundamentals, soldering, managing stress, robotics, and human factors affecting ICAM implementation. The four clinics will examine the justification of capital equipment, effective technical writing and speaking, parts and assembly cost estimating, and broaching technology.

SME also will install its 1981-82 officers at this time.

For more information contact: Tom Akas, SME Public Relations, One SME Dr., Dearborn, MI 48128, 313/271-1500.

1981 Joint Automatic Control Conference, June 17-19, 1981, U. Virginia, Charlottesville, VA. This year, the JACC will feature a session entitled "Machine Intelligence and Control in Robotics." Contact: Prof. James W. Moore, Dept. of Mech. and Aero. Eng., U. VA, Charlottesville, VA 22901, 804/924-7421.

1981 NCGA Graphics Expo. Companies holding the largest blocks of exhibit space are McDonnell-Douglas, Computervision, Synercom, Applicon, Tektronix, IBM, Gerber Systems, Autotrol Technology, Calcomp, and Technical Marketing Productions.

Conference details will be announced by the NCGA sometime in March. Approximately 50 sessions are planned, including a full day-and-a-half of tutorials. Two keynote addresses are being scheduled.

The event is being managed by the Society of Manufacturing Engineers. Inquiries concerning exhibit space should be directed to Jim McLaughlin, Exhibits Development Manager, Society of Manufacturing Engineers, One SME Dr., P. O. Box 930, Dearborn, MI 48128, 313/271-1500, ext. 309.

Previously Announced:
Refer to the original Robotics Age *Calendar announcement for details.*

Hands-On Workshop on Industrial Robot Applications, April 27-28, 1981, Atlanta, GA. Contact: Linda Propsom, Indiana-General—Magnet Products, 405 Elm St., Valparaiso, IN 46383. Announced in Vol. 3, No. 1.

National Computer Conference, 1981, Chicago, May 5-7. Contact: A. Gelles, 185 W. Houston St., NY, NY 10014. Announced in Vol. 3, No. 1.

International Conference on Computers and the Humanities, May 17-20, 1981. Contact: Richard W. Bailey, Chairman ICCH/5 Program Comm., Dept. of English, U. Michigan, Ann Arbor, MI 48109. Announced in Vol. 3, No. 1.

Automan '81, the First European Automated Manufacturing Exhibition & Conference, May 18-21, 1981, Brighton, UK. Contact: AUTOMAN '81, IFS (Conferences) Ltd., 35-39 High Street, Kempston, Bedford MK42 7BT, England. Announced in Vol. 3, No. 1.

International Conference of Data Bases in the Humanities and Social Sciences, May 20-23, 1981. Contact: Gregory A. Marks, Chairman, ICDBHSS/3 Program Committee, Institute for Social Research, U. Michigan, Ann Arbor, MI 48109. Announced in Vol. 3, No. 1.

SME Clinics, Robots: Management Overview Clinic, Cleveland, OH, June 2-4, 1981; Denver, CO, June 16-18, 1981. Contact: SME, 1 SME Dr., PO Box 930 Dearborn, MI 48128, 313/271-1500.

IEEE Computer Society Conf. on Pattern Recognition and Image Processing, Aug. 3-5, 1981, Dallas, Texas. Announced in Vol. 2, No. 3.

7th International Joint Conference on Artificial Intelligence, August 24-28, 1981, Vancouver, Canada. Announced in Vol. 2, No. 3.

BOOKS

Robotics in Practice, by Joseph F. Engelberger, forward by Isaac Asimov. AMACOM, 135 West 50th St., New York, NY 10020, $39.95.

In the preface, Engelberger states: "This book considers the place of robots in factories and it considers the types available and it makes much of economics being the driving force." Economics does play a role in **Robotics in Practice**; yet the book is well balanced, covering other aspects of the technology, such as basic robotics, methods of implementation, and even social issues.

Joseph Engelberger intended this book as a text for engineers in college, and for engineers and managers in industry who are interested in applying robots in production. And it fills its purpose admirably. To my knowledge, this is the first textbook that codifies the techniques, knowhow, and hardware needed to put robots into operation.

Those of you who read our interview with Mr. Engelberger in the last issue of *Robotics Age* (Jan/Feb, 1981) have a good idea of his background and position in Robotics. Engelberger has been working with industrial robots for twenty-five years, and he has the expertise gained by founding and running the world's oldest and largest robot manufacturer—Unimation Inc., of Danbury, Connecticut. Based on Engelberger's experience and unique viewpoint, this book is a solid, sensible introduction to the use of robots in manufacturing.

The largest section—over half of the book—is devoted to application studies, with fifteen "how to" chapters. These cover operations such as casting, welding, press work, plastics, painting, and machine loading.

There is an excellent chapter on "organizing to support robotics". It discusses how to prepare your workforce and factory floor to ease the introduction of advanced technology. The chapter also includes a reprint of an in-house report that General Electric did in 1977, called "An Organized Approach to Implementing Robots." This report alone is worth the price of the book: Capek only knows how much G.E. paid to have it done!

Most of the book, as the title indicates, is about practical robotics. However, there is a chapter on "future capabilities", and Engelberger does discuss some near-term developments, such as vision and tactile sensing. Another part of this section explains the need for robotics to stimulate the "interaction with other technologies"—the use of computers and automatic testing, for example. To reap the full benefits of automation, Engelberger asserts, the entire factory should be restructured to exploit the technology.

Despite the author's brief glance at the future, the book contains very little that is completely new. Even Engelberger's predictions are limited to the eighties. The closest the book comes to science fiction is its introduction by Isaac Asimov, a nice touch—especially since Engelberger (like all of us!) has been an Asimov admirer since his school days.

I found Engelberger's book lively and readable. The text is enriched by quite a few drawings, charts, and color photographs, and peppered with some of the author's unusual use of words—keeping the text light and, as he intended, a delight to read.

Robotics in Practice does have its flaws. The list of robot suppliers is embarrassingly short. I also would have liked the limited bibliography at the end of the book to be expanded. Still, for the first issue of a text book on a technology that changes as you write about it, it is a masterful job.

J. W. Saveriano

CLASSIFIED

HOBBYISTS: Use stepping motors in your robot for improved performance and control. New Stepping Motor Evaluation Kit makes it easy. Contains quality Airpax stepping motor, versatile driver control board and instruction book. Motor is geared down for high torque and has step angle of 0.375° for 960 steps/revolution. Completely assemble requires single 5 volt power supply. Get precise control of speed and position in your robot's mechanism. Take a "step" and order one today. Send $74.69 or Special $420.00 for 6 pcs. Check or money order to: Sparky Electronics Inc., 64-62 Dry Harbor Rd., Middle Village, New York 11379, call 212/894-3000.

ROBOTICS: Unique robot simulation on a computer. Have your own robot for $29.95. Sophisticated programming. Written in Radio Shack Level Two Basic. Send check for cassette to EMPAC, Inc., 439 Edgewood Place, Rutherford, NJ 07070.

CATALOGUE: Send stamp for descriptive catalog of books about robots and artificial intelligence. Robotic Publishing, 17095 Bridgeport Rd, Dallas, Oregon 97338.

Our classified advertising section is for readers wishing to buy, sell or trade hardware or software. The price is 30¢ per word. The first word is set in all caps. Minimum 20 words. Send copy with check to ROBOTICS AGE CLASSIFIED, P. O. Box 725, La Canada, Calif. 91011.

68000+PASCAL=
POWER

THE RSI 68K[1] PUTS ARTIFICIAL INTELLIGENCE WITHIN REACH . . .

68000 BASED RSI 68K[1]

Drastically increased processing speed and flexibility:
- 8 MHz 68000 CPU
- 32-Bit internal arithmetic registers
- 24-Bit address register
- Powerful assembly language instructions support modular programming
- Designed for expansion to 32 bit word size
- 8 Levels of interrupt priority
- Vectored interrupts and DMA fully supported
- Outbenchmarks the IBM 370/145[2] *
- High speed string processing

Dramatically increased memory:
- 68000 directly addresses 16 MB of memory
- Sorts can be done in core rather than in disk I/O

Compatibility:
- Pascal to 68000 Native-Code Compiler
- Uses DEC Q-BUS[3] and standard DEC[3] compatible peripherals
- Ability to mix a variety of CPU's in the same system. Currently supports any DEC LSI-11[3] or MOTOROLA 68000
- Forthcoming is Intel Multibus[4] Version
- Forthcoming is Ada[5] Compiler

PASCAL
- Fast program development
- Self documenting
- Supports structured programming
- Easy to update
- Easy to maintain
- Transportable from computer to computer
- Powerful logical constructs greatly simplify programming

For example: Modular procedures and functions
Strong data types
If-then-else structures
Case structures
Do-while structures
Repeat-until structures

- Local and global variables
- Recursive problem solving
- Block insert—a block of statements may be inserted anywhere one statement can exist
- Built in Boolean operations: end of file, end of line
- Library capability
- Program segmentability
- Procedure linking
- RSI[1] Pascal compiles to native 68000 code
- Built-in powerful string-handling features

*"Kilobaud Microputing" October, 1980
[1]A trademark of Renaissance Systems, Inc.
[2]A trademark of International Business Machines
[3]A trademark of Digital Equipment Corporation
[4]A trademark of Intel Corporation
[5]A trademark of Department of Defense

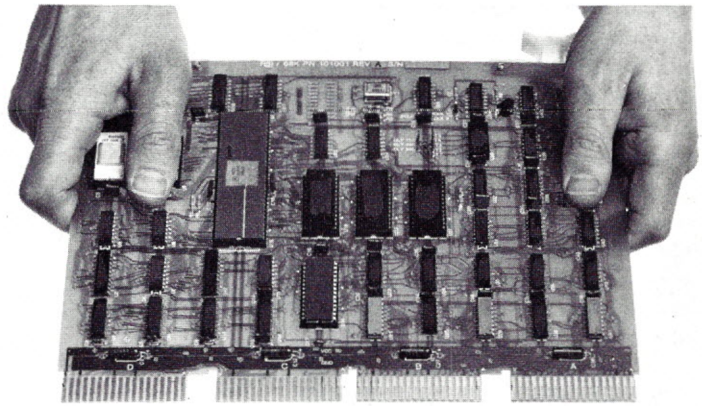
Dealerships are available.

Call us now for more information, or fill in the coupon. (301)840-0700

I AM INTERESTED! Send more information now.

Name ________________ Title ________________
Company ________________
Street ________________
City ____________ State ________ Zip ________
Phone ________________ Call me IMMEDIATELY ☐
My need is: ☐ 1-3 mo. ☐ 3-6 mo. ☐ Future project ☐ File
☐ I am interested in a dealership.

INTELLIMAC/*Intelligent Machines*
51 Monroe St., 18th Floor • Rockville, Md. 20850
Ph. (301)840-0700 • TWX: 710-828-9786 • Cable: EAII ROCKVILLE MD



R&D ENGINEERS — — —
GET YOUR HANDS ON IT!!!

QUANTUM LEAP IN PERFORMANCE OVER THE 8-BIT AND OLDER 16-BIT PROCESSORS!
Gives you overall speedup of robotic feedback cycle time.

Saves development time.
- High speed sensory data processing
- High speed string processing power
- Fast coordinate transformations
- Easy implementation of in-memory AI algorithms, predicate calculus and trajectory computations
- Design and test algorithms quicker and easier
- Both Pascal and 68000 support features that make debugging far more efficient
- Plenty of memory, no need to use extra time for "programming tricks," previously needed with limited memory
- Mixed mode listing (Pascal source statements followed by 68000 statements)

Desktop version available soon!

INTELLIMAC
Intelligent Machines

51 Monroe St., 18th Floor
Rockville, Md. 20850
(301)840-0700
TWX: 710-828-9786
CABLE: EAII ROCKVILLE MD

CIRCLE 18